

142030-9

**BİLGİSAYARLA TANK NAMLUSUNUN  
KONTROLÜ**

**Cumhur AKIN**  
**Yüksek Lisans Tezi**

**Fen Bilimleri Enstitüsü**  
**Elektrik-Elektronik Mühendisliği**  
**Anabilim Dalı**  
**Aralık - 1998**

## JÜRİ VE ENSTİTÜ ONAYI

Cumhur AKIN'ın Bilgisayarla Tank Namlusunun Kontrolü başlıklı Elektrik - Elektronik Mühendisliği Anabilim Dalındaki, Yüksek Lisans tezi 9 Aralık 1998 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Adı-Soyadı

İmza

Üye (Tez Danışmanı): Prof.Dr. Atila BARKANA

Üye : Prof.Dr. Altug İFTAR

Üye : Yard.Doç.Dr. Osman PARLAKTUNA

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun  
.....16.12.1998.....tarih ve .....19/2..... sayılı kararıyla onaylanmıştır.

## ÖZET

Yüksek Lisans Tezi

### BİLGİSAYARLA TANK NAMLUSUNUN KONTROLU

CUMHUR AKIN

Anadolu Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof.Dr.Atila BARKANA  
1998, 93 Sayfa

Bilgisayarla tank namlusunun kontrol sisteminin modernizasyonu için tasarlanmış bir projedir. Bilgisayarlı atış kontrol sistemi tank ve M52 obüslerinin namlu açısını dikey (yükseliş) ve yatay (yan) olarak istenen koordinatlara ayarlanmasını, bilgisayar kontrolü ile sağlayan bir sistemdir.

Aynı zamanda namlunun dikey (yükseliş) hareketini sağlamak için gerekli olan hidrolik denge basınç ayarını ve namlunun doğru koordinata gelmesini etkileyecek platform eğim durumunu da kontrol ederek gerekli düzeltmeleri sağlayacak yapıda donanım ve yazılım içermektedir.

Bu tezde sistemde kullanılan servo motorun dinamik denklemleri bilinmediğinden dolayı, servo motorun pratik olarak davranışları incelenmiş ve hareket eğrisinin bulunması için donanım ve yazılım hazırlanmıştır. Bunun için servo motorun transfer fonksiyonunu bulma adı altında BCX51 programı yardımıyla bir assember programı yazılmıştır.

Ayrıca sisteme ait namlunun istenen koordinatlara gelebilmesini sağlayacak olan kontrol programı BCX51 programlama diliyle yazılmıştır. Sisteme ait donanımlardan CPU kartı, DC motor sürücü kartı ve enkoder sayım kartı tasarımları gerçekleştirilmiştir.

Anahtar Kelimeler: Tank Atış Kontrolü, Dc Servo Motor, Enkoder, PI Kontrol, PWM

## **ABSTRACT**

**Master of Science Thesis**

### **TANK BARREL CONTROL WITH COMPUTER**

**CUMHUR AKIN**  
**Anadolu University**

**Graduate School of Natural and Applied Sciences**  
**Electrical and Electronics Engineering Program**

**Supervisor: Prof. Atila BARKANA**  
**1998, 93 Pages**

This masters thesis is about the modernization project of military tank shooting systems. The aim of this computer aided Shooting Control System is to adjust the horizontal and vertical angles of the tank barrel according to the given coordinates of the target.

All the required hardware and software necessary to check the slope of the platform and including those which are the pressure of the hydrolic system and then to make the suitable coordinate corrections are reported in this document. The hydrolic pressure is used to raise the barrel to a proper angle.

Since the dynamic equations of the servo motor is not available, the behaviour of the motor is investigated, and a hardware and a related software are prepared to find out the motion data. An additional assembler code is written, by using BCX51 Software, to calculate the transfer function of the servo motor.

To adjust the angles of the barrel for the given coordinates, an assembler code is also written with the help of BCX51.

**Keywords: Tank Barrel Control, Dc Servo Motor, Encoder, PI Control, PWM**

## İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET .....	i
ABSTRACT .....	ii
İÇİNDEKİLER .....	iii
ŞEKİLLER DİZİNİ .....	v
ÇİZELGELER DİZİNİ .....	vi
SİMGELER VE KISALTMALAR DİZİNİ .....	vii
1. GİRİŞ .....	1
2. SİSTEM DENKLEMLERİ .....	2
2.1. Giriş .....	2
2.2. Dc Servo Motorun Transfer Fonksiyonunun Bulunması .....	2
2.3. Dc Servo Motorlarda Elektronik Hareket Kontrolü .....	5
2.4. Dc Servo Motor Parametreleri ve Transfer Fonksiyonu .....	8
2.5. PI Kontrolcünün Tasarlanması .....	11
3. SİSTEM DONANIMI .....	16
3.1. Mikrokontrolör .....	16
3.1.1. 8051 Mikrokontrolörün İç Yapısı .....	16
3.1.2. Philips 8051 Ailesi .....	16
3.1.3. 8051 Mimarisi .....	16
3.1.4. Program Hafızası .....	17
3.1.5. Veri Hafızası .....	19
3.1.6. 80C51 İle Seri Haberleşme .....	21
3.1.7. 8051'in Eprom Ürünleri .....	22
3.2. Dc Servo Motor .....	22
3.2.1. Servo Sistem Nedir .....	22
3.2.2. Servo Sistem İle Kontrol .....	22
3.2.3. Servo Sürücü Devresi .....	23
3.3. Enkoder .....	24
4. SİSTEM YAZILIMI .....	25
4.1. Giriş .....	25
4.2. Motorun Transfer Fonksiyonunu Bulma Programı .....	26
4.3. Kontrol Programı .....	35

## İÇİNDEKİLER (DEVAM)

5. BİLGİSAYARLI ATIŞ KONTROL SİSTEMİNİN İŞLETİMİ	55
5.1. Sistemin Teknik Özellikleri	55
5.2. Sistemin Ana Elemanları	55
5.2.1. Konsol Ünitesi	55
5.2.2. Kontrol Ünitesi	55
5.3. Sistemin Çalıştırılması	57
5.4. Sistemin İşletilmesi	57
5.5. Sistemin Baiks Bağlantısı	63
5.6. Sistemin Bakımı	64
5.6.1. Eğim Ölçer Kalibrasyonu	64
5.6.2. Basınç Ölçer Kalibrasyonu	64
5.7. Sistemin Onarımı ve Arıza Tespiti	64
6. SONUÇLAR, YORUM VE ÖNERİLER	66
7. KAYNAKLAR	68
8. EKLER	69
EKLER	
Ek-1 Dc Servo Motorun Transfer Fonksiyonunu Bulma Programı	69
Ek-2 Sistem Kontrol Programı	72

## ŞEKİLLER DİZİNİ

- 2.1. (a) Armatür kontrollü dc motorun şematik gösterimi  
(b) Elde edilen denklem (2-7), (2-8) ve (2-9)'un blok diyagramı  
(c) Basitleştirilmiş blok diyagramı
- 2.2. (a) Yüksek hızlı, yüksek hassasiyet pozisyonlu servo sistemde hız kontrolü için tasarlanmış servo sürücü ve servo motor kombinasyonu  
(b) Servo sürücünün fonksiyonel diyagramı
- 2.3. Yük hareket profili
- 2.4. Pozisyon kontrollü servo motor
- 2.5. (a) Basitleştirilmiş servo sistem  
(b) Yüksek hızlı, yüksek hassasiyet pozisyonlu servo sistem
- 2.6. (a) Sistemin birim adım tepkisi  
(b) S biçimli tepki eğrisi
- 2.7. PID kontrol uygulanmış kapalı döngü sistemi
- 3.1. 8051'in blok diogramı
- 3.2. Program hafızası
- 3.3. Harici program hafıza bağlantısı
- 3.4. Harici veri hafıza bağlantısı
- 3.5. Dahili veri hafızası
- 3.6. Dahili RAM'in alt 128 baytı
- 4.1. Dc servo motorun transfer fonksiyonunu bulma programı akış diyagramı
- 4.2. Motor kontrol sistemi CPU kartı devre şeması
- 4.3. Enkoder sayım kartı devre şeması
- 4.4. Motor sürücü kartı devre şeması
- 4.5. Kontrol programı akış diyagramı
- 4.6. CPU kartı devre şeması
- 4.7. Enkoder sayım kartı devre şeması
- 4.8. PWM motor kontrol sürücü kartı devre şeması
- 5.1. Sistemin blok şeması olarak gösterimi
- 5.2. Sistemin ilk çalıştırılması sırasındaki konsol ekran görüntüsü
- 5.3. Sistemin işletilmesi sırasındaki konsol ekrandaki ikinci ana menü
- 5.4. Sistemin namlu açısı sıfırlama bölümü konsol ekran görüntüsü
- 5.5. Sistemin hareketini tamamladıktan sonraki konsol ekran görüntüsü
- 5.6. Sistemin atış bilgisi bölümü konsol ekran görüntüsü
- 5.7. Sistemin kule eğim gösterim konsol ekran görüntüsü
- 5.8. Sistemin denge basınç gösterim konsol ekran görüntüsü
- 5.9. Sistemin denge basınç ve kule eğim gösterim konsol ekran görüntüsü
- 5.10. Sistem baiks'den mesaj aldığındaki konsol ekran görüntüsü
- 5.11. Sistem kontrol kutusu üzerindeki ledlerin diziliş görüntüsü

## ÇİZELGELER DİZİNİ

- 2.1. Ziegler-Nichols kuralları
- 3.1. 8051 Ailesi
- 3.2. SCON Register bitlerinin tanımlanması
- 4.1. CPU kartı dekoder adresleme yapısı

## SİMGELER ve KISALTMALAR DİZİNİ

BAK	: Bilgisayarlı Atış Kontrol
GP	: Gerçek Basınç
$K_p$	: PI kontrolündeki oransal kazanç sabiti
$K_i$	: PI kontrolündeki integral kazanç sabiti
DG	: Değer
EGX	: X yönündeki eğim
EGY	: Y yönündeki eğim
M	: Servo motor
TURSAY	: PWM kontrolünde gerekli olan tur sayı değeri
YANL	: Yan değer düşük baytı
YANO	: Yan değer orta baytı
YANH	: Yan değer yüksek baytı
YAN	: $YANL+256*YANO+256*256*YANH$
YUKL	: Yükseliş değeri düşük baytı
YUKO	: Yükseliş değeri orta baytı
YUKH	: Yükseliş değeri yüksek baytı
YUK	: $YUKL + 256*YUKO+256*256*YUKH$
YANSON	: $YANSONL+256*YANSONO+256*256*YANSONH$
YUKSON	: $YUKSONL+256*YUKSONO+256*256*YUKSONH$
YANOFFS	: Yan offset değeri
YUKOFFS	: Yükseliş offset değeri
YANPWM	: Yan PWM değeri
YUKPWM	: Yükseliş PWM değeri
SAR	: Arıza numarası
DYAN	: Yan olarak diferansiyel gitme miktarı
DYUK	: Yükseliş olarak diferansiyel gitme miktarı
YON	: Motorun hareket yönü
PK	: Basınç eğim çarpanı
PC	: Basınç eğim sabiti
SBUF	: Seri veri tamponu
TI	: Zamanlayıcı işkesme
RI	: Register işkesme
GR	: Girilen değer
EMR	: Yapılması istenen emir
HATAL	: Yan veya yükseliş olarak yapılan hata değeri düşük baytı
HATAO	: Yan veya yükseliş olarak yapılan hata değeri orta baytı
HATAH	: Yan veya yükseliş olarak yapılan hata değeri yüksek baytı
HATA	: $HATAL+256*HATAO+256*256*HATAH$
XBY	: Harici RAM bayt okuma
YYL	: PI kontrol uygulanmış yeni yan değer düşük baytı
YYO	: PI kontrol uygulanmış yeni yan değer orta baytı
YYH	: PI kontrol uygulanmış yeni yan değer yüksek baytı
YY	: PI kontrol uygulanmış yeni yan değer

EYANHATAL	: Daha önce yan olarak yapılan hata değeri düşük baytı
EYANHATAO	: Daha önce yan olarak yapılan hata değeri orta baytı
EYANHATAH	: Daha önce yan olarak yapılan hata değeri yüksek baytı
EYANHATA	: $EYANHATAL+256*EYANHATAO+256*256*EYANHATAH$
EYUKHATAL	: Daha önce yükseliş olarak yapılan hata değeri düşük baytı
EYUKHATAO	: Daha önce yükseliş olarak yapılan hata değeri orta baytı
EYUKHATAH	: Daha önce yükseliş olarak yapılan hata değeri yüksek baytı
EYUKHATA	: $EYUKHATAL+256*EYUKHATAO+256*256*EYUKHATAH$
P	: Oransal kontrol
PI	: Oransal ve interal kontrol
PID	: Oransal, integral ve diferansiyel kontrol

## 1. GİRİŞ

Tank namlusunun hareketini mekanik sistemler yardımıyla elle kontrol etmek ve istenen açı değerine getirmek mümkündür. Ama bunu hassas bir şekilde, en kısa sürede yapmak mümkün müdür? Bu istenen şartların tam anlamıyla ve aynı zamanda sağlanabildiğini söylemek mümkün değildir. Bu yüzden; bilgisayarla tank namlusunu kontrol etme ihtiyacı duyulmuştur.

Namluyu istenen koordinatlara çok kısa sürede ve hassas bir şekilde getirmek bilgisayar sistemi ile oldukça kolaydır. Bilgisayar sistemini kullanacak personelin, manuel sistemdeki gibi özel bir eğitimden geçmesine de gerek kalmamıştır.

Gelişen teknoloji içerisinde, teknolojinin yeniliklerinden en etkin şekilde faydalanmak ihtiyaç haline gelmiştir. Bu ihtiyaçtan dolayı Kara Kuvvetleri Komutanlığı böyle bir projenin yapılmasını gerekli görmüştür.

İkinci bölümde; sisteme ait DC servo motorun dinamik denklemleri ve transfer fonksiyonu bulunmuştur. Transfer fonksiyonu bulunan servo motorun, hassas pozisyon kontrolü yapabilmesi için, servo sürücü devresi tasarlanmıştır. Ayrıca sistemdeki servo motorun daha kararlı ve hızlı işlev yapabilmesi için PI (Proportional-plus-Integral) kontrol sistemi tasarlanmıştır. PI kontrol sistemine ait olan  $K_P$  ve  $K_I$  sabitlerinin değerleri bulunmuştur.

Üçüncü bölümde; sistemde kullanılan mikrokontrolör 80C51'in genel özellikleri, programlanmasına ait bilgileri ve servo sistemler hakkında açıklayıcı bilgiler verilmiştir. Bu bilgilerin ışığı altında, tez içinde yazılmış programların anlaşılması kolaylaştırılmıştır.

Dördüncü bölümde; ilk olarak sistemde kullanılan servo motorun hareket eğrisinin, daha iyi incelenebilmesini sağlamak için yazılmış programın ve bu programa ait akış diyagramının açıklaması verilmiştir. Daha sonra ana sistemdeki, servo motorların ve diğer ünitelerin kontrol edildiği, atış kontrol programının ve bu programa ait akış diyagramının açıklamaları verilmiştir.

Beşinci bölümde; tasarlanmış olduğumuz sistemin, hangi ana bölümlerden oluştuğu ve bölümlerin kısa açıklamaları verilmiştir. Ayrıca kullanılacak olan, tasarlanmış sistemin işletilmesi hakkında görsel ve açıklamaya dayalı bilgiler verilmiştir.

Tasarlanmış olan sistem oldukça karmaşık bir yapıya sahiptir. Bu yüzden sistem içinde, DC servo motor haricinde; hidrolik sistem, sistemdeki değerlerin görüntülediği ekran ve tuş takımı, bunları kontrol eden ayrı bir mikroişlemci devresi, baiks (bilgisayarlı telsiz sistemi) haberleşmesinden, basınç ölçer ve eğim ölçer devre sistemlerinden detaylı bir şekilde bahsedilmemiştir. Bu konular hakkındaki bilgiler, gerek görüldükçe bölümler içinde verilmeye çalışılmıştır. Tez içindeki konular, programlar ve program açıklamaları genellikle, servo motor ve servo motoru kontrol eden sistemlere aittir.

## 2. SİSTEM DENKLEMLERİ

### 2.1. Giriş

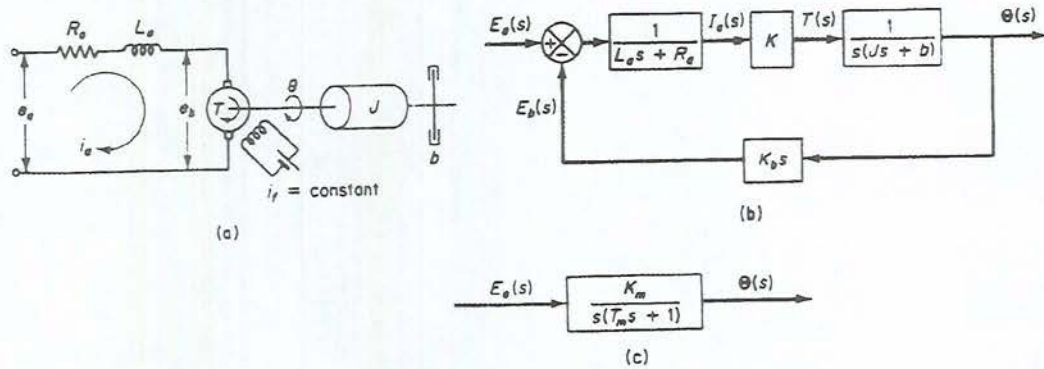
Bu bölümde, dc servo motorun pozisyon kontrolü yöntemi ile, sisteme uyarlanması için gerekli olan motor parametreleri, denklemleri ve tasarlanan PI kontrol ifadeleri yer almıştır.

Servo sistemler mekanik sistemlerde hız ve pozisyon kontrolü için tek alternatif değildir. Fakat diğer sistemlere nazaran pozisyon ve hız kontrolünde en iyi çözüm sunan sistemdir.

Bir sistem üç ana üniteden oluşur: mekanik aksam, güç besleyici amplifikatör ve motorlar, bilgisayar üniteleri. Her ana mil sisteme bir hareket verdiğinden, sistemin serbestlik derecesi eksen sayısı ile belirtilir. Her ana mile bir servo motor (doğru veya alternatif akımlı) direkt veya bir kademeli redüktör aracılığıyla bağlanır. Doğru akımlı servo motorlar hız kontrolünün kolaylığından dolayı tercih edilmekle birlikte, son zamanlarda alternatif akımlı servo motorların da hassas hız kontrolü mümkün olmaya başlamıştır. Sistemimizde doğru akım motorları ile çalışılmıştır. Doğru akım motorları sistemin hız ve moment gereksinimine göre seçilir.

### 2.2. Dc Servo Motorun Transfer Fonksiyonunun Bulunması

Herhangi dc servo motorun transfer fonksiyonu, aşağıda belirtilen parametreler yardımıyla ifade edilir: [9]



Şekil 2.1: (a) Armatür kontrollü dc motorun şematik gösterimi (b) Elde edilen denklem (2-7),(2-8) ve (2-9)'un blok diyagramı (c) Basitleştirilmiş blok diyagram

$R_a$ : Armatür direnci, ohm

$L_a$ : Armatür endüktansı, henry

$i_a$ : Armatür akımı, amper

$i_f$ : Alan akımı, amper

$e_a$ : Uygulanan armatür voltajı, volt

$e_b$ : Geri emf, volt

$\theta$ : Motor şaftının açısal yer değişimi, radyan

$T$ : Motorun ürettiği tork, N-m

J : Motorun atalet momenti, kg-m<sup>2</sup>  
 b : Motorun viskoz sürtünme katsayısı, N-m/rad/s

Motorun ürettiği tork(T); motor sabiti, armatür akımı ve hava aralığı akı miktarının çarpımı ile orantılıdır. Hava aralığı akı ifadesi

$$\psi = K_f i_f \quad (2-1)$$

şeklindedir. Burada  $K_f$  alan akım sabiti,  $i_f$  alan akımıdır. Bu durumda yeni tork ifadesi

$$T = K_f i_f K_1 i_a \quad (2-2)$$

olarak bulunur. Burada  $i_f$  alan akımı sabittir.  $i_f$  değeri,  $K_f$  alan akımı sabiti ile  $K_1$  sabitinin çarpımını kısaca K motor-tork sabiti olarak ifade edebiliriz. Bu durumda

$$T = K i_a \quad (2-3)$$

olur. Torkun işareti, armatür akımının işaretine bağlıdır. Armatür akımının işareti ise rotorun dönüş yönüne bağlıdır.

Armatür döndüğü zaman, armatürde üretilen açısız hızın ve akımın çarpımı sonucunda bu değerlerle orantılı bir gerilim oluşur. Sabit akı değeri için, üretilen gerilim ( $e_b$ )

$$e_b = K_b \frac{d\theta}{dt} \quad (2-4)$$

şeklinde ifade edilir. Bu formülde  $e_b$  geri emf ve  $K_b$  geri emf sabiti olarak isimlendirilir.

Armatür kontrollü dc servo motorların hızı, armatür voltajı  $e_a$  ile kontrol edilmektedir. (Güç katının çıkışı armatür voltajıdır.) Bu durumda armatür devresi için aşağıdaki diferansiyel denklem yazılabilir.

$$L_a \frac{di_a}{dt} + R_a i_a + e_b = e_a \quad (2-5)$$

Motorda oluşan moment, sistem mil düzeneğindeki ataleti (J) ivmelendirmek ve motordaki viskoz sürtünmeyi yenmek için harcanır. Bu durumda

$$J \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = T = K i_a \quad (2-6)$$

denklemini yazılabilir. Bütün başlangıç koşullarının sıfır olması şartı altında elde edilen denklemlerin Laplace dönüşümü alınır

$$K_b s \Theta(s) = E_b(s) \quad (2-7)$$

$$(L_a s + R_a)I_a(s) + E_b(s) = E_a(s) \quad (2-8)$$

$$(Js^2 + bs) \Theta(s) = T(s) = KI_a(s) \quad (2-9)$$

elde edilir.  $E_a(s)$  değerini giriş,  $\Theta(s)$  değerini çıkış olarak ifade ederek transfer fonksiyonu denklemini aşağıdaki şekilde yazarız.

$$\frac{\Theta(s)}{E_a(s)} = \frac{K}{s[L_a Js^2 + (L_a b + R_a J)s + R_a b + KK_b]} \quad (2-10)$$

Endüktans  $L_a$ , armatür devresinde genellikle küçüktür ve ihmal edilir. Eğer  $L_a$  ihmal edilirse, Denklem (2-10)'da verilen transfer fonksiyonu Denklem (2-11)'deki basit hale dönüşür.

$$\frac{\Theta(s)}{E_a(s)} = \frac{K_m}{s(T_m s + 1)} \quad (2-11)$$

Denklem (2-11)'de  $K_m$  motor kazanç sabiti,  $T_m$  motor zaman sabiti olarak ifade edilir. Bu sabitlerin açık ifadeleri Denklem (2-12) ve (2-13)'de verilmiştir.

$$K_m = \frac{K}{(R_a b + KK_b)} \quad (2-12)$$

$$T_m = \frac{R_a J}{(R_a b + KK_b)} \quad (2-13)$$

Armatür kontrollü dc motor sistemin durum uzayı modellemesini yapılırsa, ilk olarak Denklem (2-11)'deki diferansiyel denklemden faydalanarak;

$$\ddot{\theta} + \frac{1}{T_m} \dot{\theta} = \frac{K_m}{T_m} e_a \quad (2-14)$$

$x_1$  ve  $x_2$  durum değişkenleri,

$$x_1 = \theta \quad (2-15)$$

$$x_2 = \dot{\theta} \quad (2-16)$$

$u$  giriş değişkeni,

$$u = e_a \quad (2-17)$$

ve  $y$  çıkış değişkeni,

$$y = \theta = x_1 \quad (2-18)$$

olarak tanımlanır. Verilen dc motor sisteminin, durum uzayı gösterimi Denklem (2-19) ve (2-20)'de gösterildiği şekildedir:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T_m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_m}{T_m} \end{bmatrix} u \quad (2-19)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2-20)$$

### 2.3. Dc Servo Motorlarda Elektronik Hareket Kontrolü

Servo motorlar için çok farklı tipte elektronik hareket kontrolcü veya servo sürücü mevcuttur. Dc servo motorların hız kontrolü için çoğunlukla servo sürücüler tasarlanır. Şekil 2.2.'de yüksek hızlı, yüksek hassasiyet pozisyonlu servo sistemde hız kontrolü için tasarlanmış servo sürücü ve servo motor kombinasyonu gösterilmiştir.

Denklem (2-11)'de  $u$  (Volt) şiddetinde bir adım fonksiyonu verildiğinde  $\left(\frac{u}{s}\right)$  ve sistemin ters Laplace transformu hesaplandığında motorun hızı, Denklem (2-21)'de gösterildiği şekilde değişir. [5]

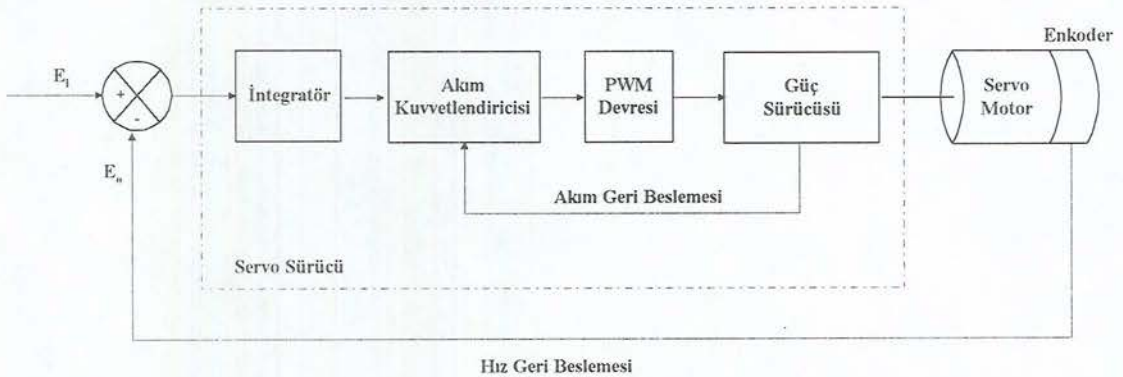
$$w(t) = K_m u \left[ 1 - \exp\left(-\frac{t}{T_m}\right) \right] \quad (2-21)$$

Denklem (2-21)'e göre hızda hiç bir zaman salınım olmaz. Gerçekte sistemin diferansiyel denklem derecesi çok daha yüksek olduğu için salınımlar olabilir. Fakat sistemde hız devresi salınım olmayacak şekilde ayarlanır. Motor sabiti ( $K_t$ ) ile tako-metre sabiti ( $H_g$ ) imalatçı tarafından verilir.  $R_g$  ve  $T_g$  kazançları servo amplifikatörün-deki tornavida girişli trimpotlar aracılığıyla ayarlanır. Bu ilk servo ayarlama sırasında pozisyon kontrolü tamamen devreden çıkartılır. Pratikte bir fonksiyon üreticisi ile ortalaması sıfır olan, +5 V ile -5 V arasında 10 Hz frekanslı bir kare dalga ( $u$ ), servo amplifikatörün girişine bağlanır. Servo motorun hızı ( $\omega$ ) iki girişli bir osiloskopta gözlenir. Girişlerden birisi kare dalga sinyal, diğeri ise motorun hızı ile doğru orantılı olan takometre sinyalidir. Motor 10 Hz'de (50 milisaniye sağa 50 milisaniye sola dönerek) salındıkça, motorun mili ileri geri hareket eder. Eğer sabit bir voltaj verilirse motor aynı yönde dönerek hızlanır istek dışı hareket eder. Bu tehlikeli durumu önlemek için ortalaması sıfır olan kare sinyal girişi uygulanır. Sinüs eğrisi verilirse sistemin dinamiği tahrik olmayacağı için servo sistemin ayarı imkansızdır. Pratikte servo ayarı sırasıyla şu şekilde yapılır: Önce  $R_g$  kazancı tornavida ile minimuma (ileri kazanç), sonra takometre kazancı  $T_g$  (geri besleme

kazancı) maksimuma getirilir. Bu-nun sebebi takometrenin düşük hızlarda verdiği voltajın kablolardaki akım gürültüsüne baskın olması içindir. Eğer motorun hızı, kare dalganın köşesine yüksek frekansta salınım yapmadan, düzgün bir şekilde yaklaşırsa ileri kazanç ( $R_g$ ) trimpotu tornavida ile yavaş yavaş hızın salınım yapıp istenilen seviyeyi aşmaya başlamasına kadar artırılır. Burada amaç, ileri kazancı artırarak sistemin istenen hız daha kısa bir zamanda ivmelenip çıkabilmesidir. Sistemin kare sinyaline hiç salınım yapmadan düzgün şekilde yaklaşması ileri kazancın bu limit noktasından, hafifçe geri çekilmesiyle sağlanır. Sistemin bir ana mil servosu ayarlandıktan sonra, aynı kare sinyal sırasıyla diğer eksen kontrol servolarına da uygulanır. Dikkat edilecek husus, tüm servoların hız kontrol eğrisinin, yani transfer fonksiyonlarının hassas bir şekilde eşitlenmesidir. Servo eşitleme bir eksen referans kabul edilerek yapılır. Aynı kare sinyali referans ve eşitlenecek servo güç besleyicisine verilir. Eşitlenen servonun ileri kazanç trimpotu ayarlanarak servoların kare dalga tahrikine olan cevapların aynı olması sağlanır. Eğer iki eksen servoları dinamik olarak eşitlenemez ise sistemdeki doğrusal veya dairesel interpolasyonlar, komut konumlarından sapmalar doğurur ve istenen koordinata ulaşmaktaki hassasiyet kaybolur. [6]



(a)



(b)

Şekil 2.2: (a) yüksek hızlı, yüksek hassasiyet pozisyonlu servo sistemde hız kontrolü için tasarlanmış servo sürücü ve servo motor kombinasyonu (b) servo sürücünün fonksiyonel diyagramı

Eksen servolarının ayarlanıp eşitlenmesinden sonra, geri beslemeli pozisyon kontrol devresi, enkoder ve bilgisayar ünitesinin bağlanması ile harekete geçirilir. Pozisyon kontrolü için ayarlanacak yegane sistem dijital kontrol veya filtre ( $D(z) = K_p \frac{(z-a)}{(z-b)}$ ) parametrelerinin sayısal değerlerinin hesaplanmasıdır. Bu değerler gerçekte iyi bir kontrol analizi ve bilgisayar simülasyonu ile yapılır. Fakat endüstride çalışan mühendisler aşağıdaki basit tekniği kullanarak dijital pozisyon kontrol devresini ayarlayabilirler. Geri beslemeli sistemin kesikli zamandaki açık döngü transfer fonksiyonu, enkoder kazancını ve pozisyon hatalarının integralini içeren dijital sayıcıyı, hız kontrol devresine ilave ederek bulunur.

$$G(s) = \frac{\Theta(s)}{E_a(s)} = \frac{K_m}{s(T_m s + 1)} \quad (2-22)$$

Denklem (2-22)'deki sürekli zamandaki sistem, bilgisayardan dijital/analog (D/A) devresiyle, sabit kontrol zaman (T) aralıkları ile iletilen dijital komutlarla tahrik edilir. Dolayısıyla sistemin kesikli zamandaki karşılığı (Zero Order Hold Equivalent) bulunur.

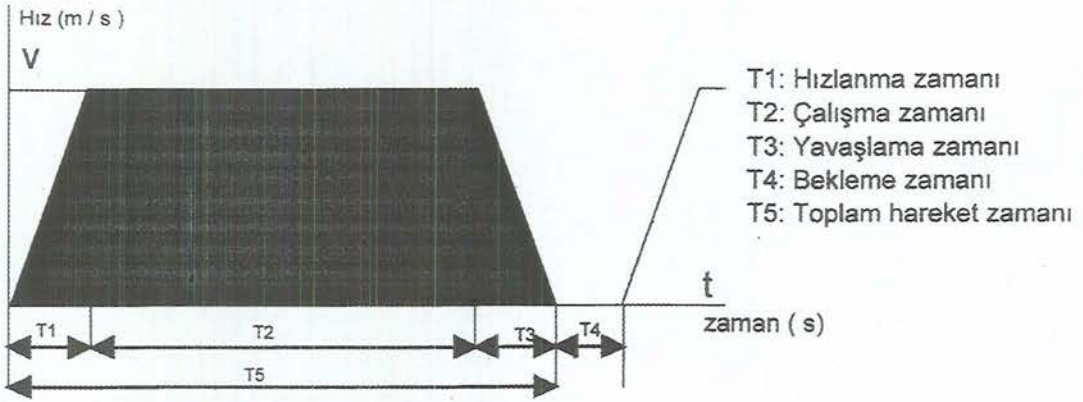
$$G(z) = \frac{D(z)G(z)}{1 + D(z)G(z)} = K_m K_p \left[ \frac{T(z - e^{-T/T_m}) - T_m(1 - e^{-T/T_m})(z - 1)}{z(z - 1)(z - e^{-T/T_m})} \right] \quad (2-23)$$

Burada z ileri fark (forward difference) operatörü olarak anılır. Açık devre transfer fonksiyonunun kutbu ( $z - e^{-T/T_m}$ ) dijital filtrenin sıfırı a ile eşitlenip, dijital filtrenin kutbuna sıfır değeri verilir. (b=0) Bu arada filtrenin kazancı  $K_p=1$  olarak tutulur. Sistemde ileri geri kumandalar verilip takometreden ölçülen hızda titreşim olup olmadığına bakılır.  $K_p$  titreşim sınırının alt limitine kadar arttırılır.  $K_p$  kazancının yüksek olması servonun istenen hızlara daha çabuk ulaşmasını sağlar. Sonuçta geri beslemeli ilerleme mili pozisyon kontrol servosunun kesikli zamandaki transfer fonksiyonu,

$$G(z) = \frac{D(z)G(z)}{1 + D(z)G(z)} = K_m K_p \left[ \frac{T(z - e^{-T/T_m}) - T_m(1 - e^{-T/T_m})(z - 1)}{z(z - 1)} \right] \quad (2-24)$$

Pratikte en kolay hız kontrolü sabit ivmelendirme hız profili, (Trapez Hız Profili) ile yapılır. (Şekil 2-3) İvmelendirme (T1) ve yavaşlama hızları (T3) ilerleme mili tertibatının ataletine bağlıdır. Sistemin değişken hızı bilgisayar saatinin frekansını program ile ayarlayarak yapılır.

Entegrasyon zaman aralığı  $T_i$  ivmelendirme süresince (T1) gittikçe kısaltılır, sabit hızda (T2) sabit tutulur, yavaşlamada (T3) gittikçe uzatılır. Sistem tasarımında enterpolasyon zaman aralıkları pozisyon kontrol zaman dilimine eşittir.



Şekil 2-3: Yük hareket profili

#### 2.4. Dc Servo Motor Parametreleri ve Transfer Fonksiyonu

Sistemde kullanılan dc servo motora ait parametreler aşağıdaki şekildedir:

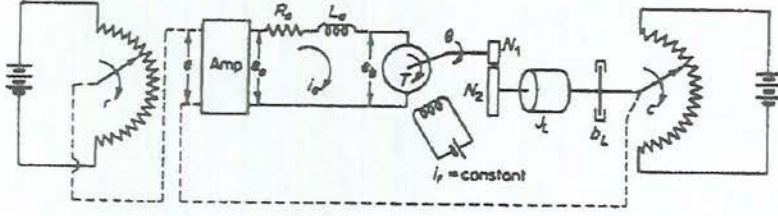
- $r$  : Referans giriş milinin açısal yer değiştirmesi, radyan
- $c$  : Çıkış milinin açısal yer değiştirmesi, radyan
- $\theta$  : Motor milinin açısal yer değiştirmesi, radyan
- $K_1$  : Potansiyometrik hata dedektörünün kazancı= $24/\pi$  volt/rad
- $K_p$  : Kuvvetlendirici kazancı= $10$  Volt/volt
- $e_a$  : Uygulanan armatür gerilimi, volt
- $e_b$  : Geri emf, volt
- $R_a$  : Armatür sargı direnci= $0.2$  ohm
- $L_a$  : Armatür sargı endüktansı=ihmal edilebilir
- $i_a$  : Armatür sargı akımı, amper
- $K_b$  : Geri emf sabiti= $0.21$  volt-sn/rad
- $K$  : Motor tork sabiti= $0.055$  N-m/amp.
- $J_m$  : Motorun atalet momenti= $6 \times 10^{-3}$  kg-m<sup>2</sup>
- $b_m$  : Motorun vizkos sürtünme katsayısı=ihmal edilebilir
- $J_L$  : Yükün atalet momenti= $4.4 \times 10^{-2}$  kg-m<sup>2</sup>
- $b_L$  : Yükün vizkos sürtünme katsayısı= $2.3 \times 10^{-2}$  N-m/rad/sn
- $n$  : Redüktör oranı  $N_1/N_2=1/10$

Servo motora ait verilen yukarıdaki parametre değerlerini kullanarak sistemin kapalı döngü transfer fonksiyonu bulunur:

$$E(s)=K_1(R(s)-C(s))=7.64(R(s)-C(s)) \quad (2-25)$$

Kuvvetlendirici için;

$$E_a(s)=K_p E(s)=10E(s) \quad (2-26)$$



Şekil 2.4: Pozisyon kontrollü servo motor

Armatür kontrollü dc motorun eşdeğer atalet momenti  $J$  ve eşdeğer vizkos sürtünme sabiti  $b$ 'yi hesaplayalım:

$$J = J_m + n^2 J_L$$

$$J = 6 \times 10^{-3} + (1/10)^2 4.4 \times 10^{-2}$$

$$J = 64.4 \times 10^{-4}$$

$$b = b_m + n^2 b_L$$

$$b = 2.3 \times 10^{-4}$$

Denklem (2-12) ve (2-13)'den faydalanarak;

$$\frac{\Theta(s)}{E_a(s)} = \frac{K_m}{s(T_m s + 1)}$$

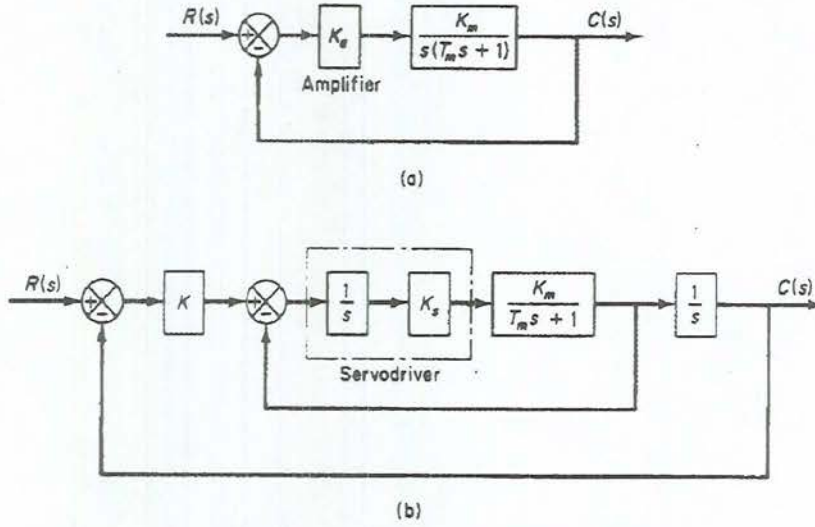
$$K_m = \frac{K}{(R_a b + K K_b)} = \frac{55 \times 10^{-3}}{(0.2) \times (2.3 \times 10^{-4}) + (55 \times 10^{-3}) \times (21 \times 10^{-2})} = 4.743$$

$$T_m = \frac{R_a J}{(R_a b + K K_b)} = \frac{(0.2) \times (64.4 \times 10^{-4})}{(0.2) \times (2.3 \times 10^{-4}) + (55 \times 10^{-3}) \times (21 \times 10^{-2})} = 0.111$$

Elde edilen sonuçlara göre;

$$\frac{\Theta(s)}{E_a(s)} = \frac{10C(s)}{E_a(s)} = \frac{K_m}{s(T_m s + 1)} = \frac{4.743}{s(0.111s + 1)} \quad (2-27)$$

Denklem (2-27)'de açık döngü sistemin transfer fonksiyonunu bulmuş oluyoruz. Şekil 2.5'de servo sürücülü, pozisyon geribeslemeli kapalı döngü sistemi verilmiştir:



Şekil 2.5: (a) Basitleştirilmiş servo sistem (b) yüksek hızlı, yüksek hassasiyet pozis-yonlu servo sistem

Şekil 2.5'de verilen kapalı döngü sistemin adım adım transfer fonksiyonunu bulalım:

$$G_1(s) = K_s \frac{1}{s} \frac{K_m}{T_m s + 1} = \frac{K_s K_m}{s(T_m s + 1)} \quad (2-28)$$

$$G_2(s) = \frac{G_1(s)}{1 + G_1(s)} = \frac{\frac{K_s K_m}{s(T_m s + 1)}}{1 + \frac{K_s K_m}{s(T_m s + 1)}} = \frac{K_s K_m}{s(T_m s + 1) + K_s K_m} \quad (2-29)$$

$$G_3(s) = K \frac{1}{s} G_2(s) = K \frac{1}{s} \frac{K_s K_m}{s(T_m s + 1) + K_s K_m} = \frac{K K_s K_m}{T_m s^3 + s^2 + K_s K_m s} \quad (2-30)$$

$$G(s) = \frac{G_3(s)}{1 + G_3(s)} = \frac{K K_s K_m}{T_m s^3 + s^2 + K_s K_m s + K K_s K_m} \quad (2-31)$$

## 2.5. PI Kontrolcünün Tasarlanması

Bazı proseslerin doğru transfer fonksiyon modellerinin hazırlanmasındaki zorluktan dolayı, kontrolörler için optimum kazanç değerlerinin hesaplanmasında kullanılan, iz gösterici deneysel bazlı çalışmalar yapılmıştır. En çok kullanılan yöntem Ziegler-Nichols kurallarıdır ki, bulunmalarından bu yana geçen 40 yıllık süre içerisinde bile hala kullanılmaktadırlar. [7] ve [9]

Kurallar uygulamada iki metoda ayrılır. Birinci metot, sistemin açık-döngü adım tepkisine ihtiyaç duyarken, ikinci kural kontrolörün yüklediği, yapılmış olan deney sonuçlarını kullanır. Her ne kadar kurallar analitik modeli olmayan sistemlerin tasarlanmasında kullanılması gibi birincil amaca sahipken, ayrıca bir model tasarlanırken dahi kullanılabilir.

Ziegler ve Nichols kurallarını, deneylerden ve bir çok değişik endüstriyel uygulamaların sonucunda elde etmişlerdir. IAE kriterini birim-adım tepkisi ile kullanırken, kontrolörlerin aşağıda gösterilen ve osilasyonlu bir adım tepkisine fakat yeterli salınım sayesinde ikinci tepe noktasının ilk tepe noktasına göre %25 aşığınsın kurallara bağlı olarak ayarlandığını bulmuşlardır. Bu çeyrek-düşüş kriteridir ve bazen de belirleyici özellik olarak kullanılabilir.

Birinci metot, işlem reaksiyon metodu, Şekil 2.6'da gösterilen açık-döngü adım tepkisine sahip sistemlere dayandırılır. Bu işlem davranışı "R" ve "L" gibi iki parametre tarafından karakterize edilmiştir. "R" tepki eğrisinin en dik parçasına olan eğim çizgisinin tanjantı ve "L" bu çizginin zaman eksenini kestiği noktadaki zamandır. Birinci ve ikinci dereceli lineer sistemler "L" nin pozitif değerleri için sonuç vermezler ve böylece metot bu tür sistemlere uygulanamaz. Bununla birlikte, yeterli salınıma sahip üçüncü derece veya daha yüksek lineer sistemler buna benzer bir tepki sonucu verirler. Böylelikle, Ziegler-Nichols kuralları Tablo 2.1'de verilen kontrolör ayarlama değerlerini isterler.

Son-devir metodu, sistemde kontrolörün bulunduğu deneyleri kullanır. Oransal kazanç  $K_p$ 'nin düşük değere ayarlanması ile işleme başlanır ve oransal mod haricinde diğer kontrol modları kapatılır. Kazanç, sistemde güçlü osilasyonlar başlayana kadar yavaşça artırılır. Bu osilasyon periyodu  $P_u$  ve bununla ilgili en son kazanç değeri  $K_{pu}$  olarak adlandırılır. Ziegler-Nichols'un bu parametrelere uygun olarak hazırlanmış oldukları tablosu Tablo 2.1'de verilmiştir. Oransal kazanç değeri, "P" kontrolüne göre "PI" kontrolü için düşük, "PID" kontrolü için yüksektir, çünkü "I" (integrasyon) işlemi siste-

min derecesine göre artar. Böylece sistemin tepkisini bozmaya çalışır; bu durumda düşük bir kazanç ihtiyacı duyulur. Diğer taraftan “D” (diferansiyel) işlemi sistemi sabitlemeye çalışır, bu nedenle oransal kazanç sabit karakteristiklerde derecelendirilmeden arttırılır. Kurallar bir çok tipteki işlemlerin tipik durumundan icat edildiği için, sistemdeki son ayarlama çoğu zaman mecburi bir gerekliliktir. Aşağıda yapılan işlemler son-devir metodunun analitik bir modele nasıl uygulanabileceğini göstermektedir.

Tablo 2.1: Ziegler-Nichols kuralları

Kontrolcü transfer fonksiyonu:  $G(s) = K_p \left( 1 + \frac{1}{T_I s} + T_D s \right) = K_p + \frac{K_I}{s} + K_D s$

Kontrol Modu	İşlem Reaksiyon Methodu	Son Devir Metodu
P kontrolü	$K_p = 1 / RL$	$K_p = 0.5 K_{pu}$
PI kontrolü	$K_p = 0.9 / RL$ $T_I = 3.3 L$	$K_p = 0.45 K_{pu}$ $T_I = 0.83 P_u$
PID kontrolü	$K_p = 1.2 / RL$ $T_I = 2 L$ $T_D = 0.5 L$	$K_p = 0.6 K_{pu}$ $T_I = 0.5 P_u$ $T_D = 0.125 P_u$

Hareketli paya sahip olmayan oransal kontrollü üçüncü derece sistemler, aşağıdaki karakteristik eşitliğin çıkarılmasında yol gösterirler:

$$a_3 s^3 + a_2 s^2 + a_1 s + a_0 + K_p = 0 \quad (2-32)$$

eşitlikte  $K_p$  oransal kazançtır. Son periyot  $P_u$  ve buna bağlı olan oransal kazanç  $K_{pu}$  bulunur. Sistemde baştan sona kadar aynı seviyede süren osilasyon, bir çift kökün tamamen sanal olduğunu ve diğer köklerin negatif gerçek parçalara sahip olduğunu gösterir. Bunun ne zaman oluştuğunu bulabilmek için,  $\omega_u$  osilasyonun henüz bilinmeyen bir frekansı belirttiği için  $s = i\omega_u$  eşitliğini sisteme uygulamak gerekir. Bu durumda  $s^2 = -\omega_u^2$  ve  $s^3 = -i\omega_u^3$  olacaktır. Bu elde edilen formülleri Denklem (2-32)'deki yerlerine koyduktan ve gerçek ve sanal kısımları toplandıktan sonra, aşağıdaki eşitliği elde ederiz:

$$(a_0 + K_{pu} - a_2\omega_u^2) + i(a_1\omega_u - a_3\omega_u^3) = 0 \quad (2-33)$$

Hem gerçekte hem de sanal terimler sıfır olmalıdır, böylece denklemede bilinmeyen  $\omega_u$  ve  $K_{pu}$ 'nin bulunması için iki eşitlik elde ederiz.

$$a_0 + K_{pu} - a_2\omega_u^2 = 0 \quad (2-34)$$

$$\omega_u(a_1 - a_3\omega_u^2) = 0 \quad (2-35)$$

Denklem (2-35)'deki eşitlikte  $\omega_u = 0$  olması mümkün değildir. Bu yüzden,

$$\omega_u = \pm \sqrt{\frac{a_1}{a_3}} \quad (2-36)$$

Elde edilen Denklem (2-36)'deki eşitliği Denklem (2-34)'deki eşitlikte yerine koyarsak, son periyot oransal kazancının şu formülden elde edildiğini görürüz.

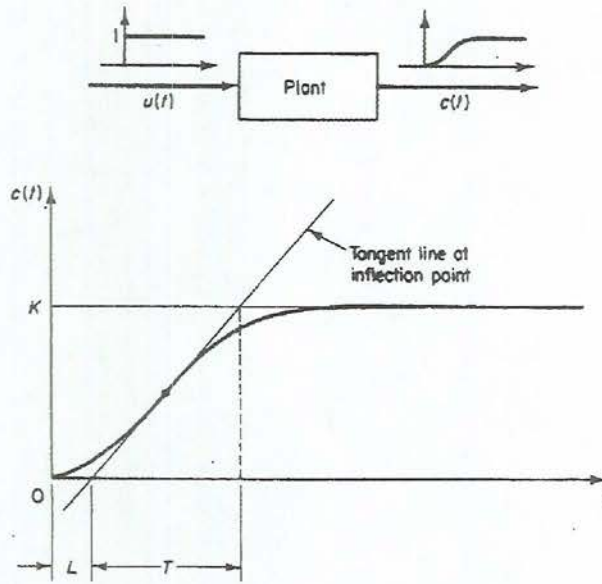
$$K_{pu} = a_2\omega_u^2 - a_0 = \frac{a_2a_1}{a_3} - a_0 \quad (2-37)$$

Son periyotta şöyle olacaktır,

$$P_u = \frac{2\pi}{\omega_u} \quad (2-38)$$

Denklem (2-32)'nin çözümünden köklerden birisinin  $s = -a_2/a_3$  olacağı görülür. Eğer  $s < 0$  ise, Denklem (2-37) ve Denklem (2-38)'de verilen değerler Tablo 2.1'de optimum kazancı hesaplamakta kullanılabilir. Unutmayınız ki bu analizler sadece oransal kontrolü kullanır, fakat diğer kontrol kuralları için kazançlar bu sonuçlarla hesaplanabilir. Metodun güçlülüğü Denklem (2-32)'de de görüldüğü üzere köklerinin üçüncü derecesini çözmediğimizden de bellidir. Bu işlemler uygulanmış bir sistem eğer Ziegler-Nichols kuralları ile çözümlenmezse, dördüncü derece karakteristik eşitlikleri ile karşılaşılabilecektir.

Bir çok sistem için,  $K_{pu}$ 'nin değeri kararsız sistem oluşturmadan elde edilen maksimum kazancı belirtir.



Şekil 2.6: (a) Sistemin birim adım tepkisi (b) S biçimli tepki eğrisi

Aşağıda sisteme ait transfer fonksiyonu verilmiştir. Bu transfer fonksiyonuna göre bir PI kontrolcü tasarımı yapalım:

$$G(s) = \frac{G_3(s)}{1 + G_3(s)} = \frac{KK_s K_m}{T_m s^3 + s^2 + K_s K_m s + KK_s K_m}$$

Üçüncü derece sistemin karakteristik denklem katsayılarını aşağıdaki şekilde belirlenmiştir:

$$a_0 = K K_s K_m$$

$$a_1 = K_s K_m$$

$$a_2 = 1$$

$$a_3 = T_m$$

Dc servo motora ait diğer gerekli parametreler aşağıdaki şekilde belirlenmiştir:

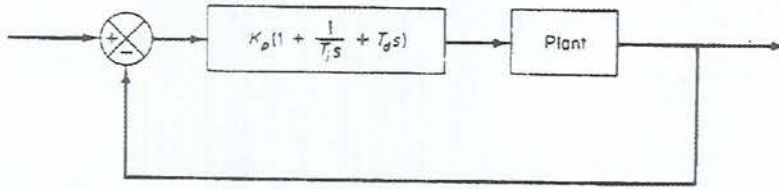
$K_s$ : Akım güçlendirici kazancı = 4.312 A/V

$K$ : Komut sinyal kazancı = 0.214 V/V

$K_m$ : Motor kazanç sabiti = 4.743 (Denklem (2-27)'de elde edilen)

$T_m$ : Motor zaman sabiti = 0.111 (Denklem (2-27)'de elde edilen)

Verilen parametrelerden faydalanarak, Ziegler-Nichols kurallarını uygulayarak PI kontrolcüye ait  $K_p$  ve  $K_I$  katsayı değerlerini hesaplayalım:



Şekil 2.7: PID kontrol uygulanmış kapalı döngü sistemi

$$K_{pu} = a_2 \omega_u^2 - a_0 = \frac{a_2 a_1}{a_3} - a_0 = (1 \times K_s K_m) / T_m - K K_s K_m$$

$$K_{pu} = (4.312 \times 4.743) / 0.111 - 0.214 \times 4.312 \times 4.743 = 179.87$$

$$\omega_u = \pm \sqrt{\frac{a_1}{a_3}} = (K_s K_m / T_m)^{1/2} = (184.25)^{1/2} = 13.573$$

$$P_u = \frac{2\pi}{\omega_u} = 2\pi / 13.573 = 0.463$$

Tablo 2-1'den faydalanarak PI kontrol için verilen değerleri kullanalım:

$$K_p = 0.45 K_{pu} = 0.45 \times 179.87 = 80.9415$$

$$T_I = 0.83 P_u = 0.83 \times 0.463 = 0.384$$

$$K_I = K_p / T_I = 80.9415 / 0.384 = 210.785$$

Bulmuş olduğumuz PI kontrolcüye ait  $K_p$  ve  $K_I$  katsayı değerleri sistem için optimum olabilecek katsayı değerleridir. Sistemde bu katsayılar, optimum değerlerinden faydalanarak belirlenir. Denemeler sonucunda, sistemdeki katsayılar  $K_p = 255$  ve  $K_I = 255$  olarak belirlendi. Bu katsayı değerleri altında sistem en iyi performansta ve gürültüsüz çalışmaktadır.

### 3. SİSTEM DONANIMI

#### 3.1. Mikrokontrolör

Mikrokontrolör ismi günümüz dünyasında , mikroişlemci ve mikrobilgisayar ismiyle birlikte anılmaktadır. Bunlar arasındaki farkları açıkça ortaya koymak, mikrokontrolörlerin önemini anlamak açısından yararlı olacaktır. [1]

Bir mikroişlemci, bir bilgisayarın CPU (Central Processing Unit) kısmıdır. Tam bir sistem için gerekli olan hafıza, I/O ve çevre elemanları içermez.

Bir mikroişlemci, hafıza, I/O ve diğer çevre elemanları ile donatılırsa mikrobilgisayar halini alır. Eğer bu donatma çip düzeyinde olursa tek çip mikrobilgisayarlar ortaya çıkarlar.

Tek çip mikrobilgisayarlar, endüstriyel kontrol uygulamalarında kullanılmak üzere tasarlanırlarsa tek çip mikrokontrolör adını alırlar. Temelde mikrobilgisayarlar ile mikrokontrolörler arasında fark yoktur. Bunlar kullanıldıkları yere göre isim alırlar.

##### 3.1.1. 8051 Mikrokontrolör İç Yapısı

8051 piyasada oldukça sık kullanılan bir mikrokontrolördür. İçinde iki adet 16 bit'lik sayıcı, 4 Kbyte kapasiteli sadece okunabilir (ROM = Read Only Memory) bir program hafızası, 128 Byte kapasiteli veri hafızası (RAM=Random Access Memory), osilatör ve programlanabilen dört paralel ve bir seri haberleşme portu bulunmaktadır. [1]

##### 3.1.2. Philips 8051 Ailesi

Philips 8051, klasik bir mikrokontrolör ve aynı zamanda paralel ve seri I/O, sayıcı/zamanlayıcı, RAM, EPROM ve rom içeren bir mikrobilgisayardır. 8051 ailesinin üyeleri ve bazı özellikleri Çizelge 3.1'de verilmiştir.

Çizelge 3.1: 8051 Ailesi

Cihaz Adı	ROMsuz Sürümü	EPROM Sürümü	ROM (Byte)	RAM (Byte)	16 Bit Zamanlayıcı
8051	8031	8751	4K	128	2
8051 AH	8031 AH	8751 H	4K	128	2
8052 AH	8032 AH	8752 BH	8K	256	3

##### 3.1.3. 8051 Mimarisi

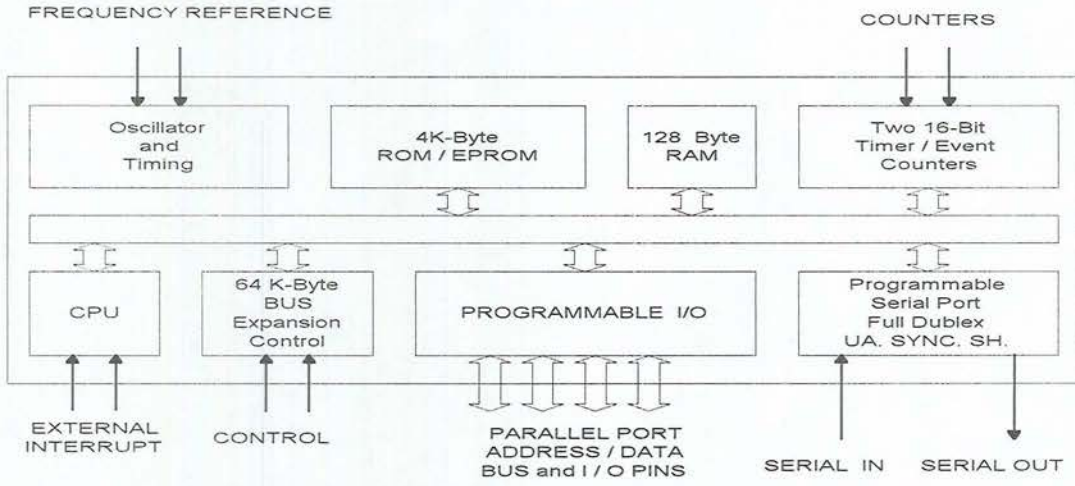
Tüm 8051 ailesi üyeleri, program ve veri hafızaları için ayrı adres aralıkları kullanılır. Program ve veri hafızasının mantıksal ayrımı, veri hafızasına 8 bitlik bir adres hattı ile ulaşılmasını sağlar. [1]

Program hafızası sadece okunabilir özelliktedir. (ROM veya EPROM) Bu hafıza 64Kbyte'a kadar çıkabilir. 8051 de en az 4Kbyte program hafızası çip üzerinde bulunur.

Bunun yanında, ROM bulunmayan sürümlerde ise tüm program hafızası harici olarak kullanılır.

Veri hafızası (RAM) program hafızasından farklı bir adres uzayı işgal eder, 8051'de 128 byte veri hafızası çip üzerindedir. Bu sayı 64 Kbyte'a kadar çıkabilir.

Harici program ve veri hafızaları kullanılması durumunda  $\overline{PSEN}$  (Program Store Enable),  $\overline{RD}$  (Read),  $\overline{WR}$  (Write) sinyalleri kullanılmaktadır.



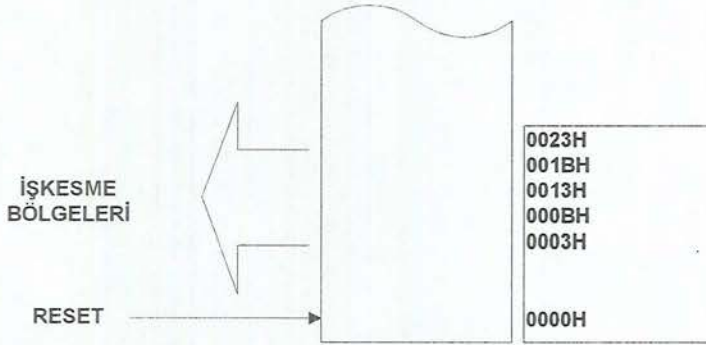
Şekil 3.1 : 8051'in blok diagramı

### 3.1.4. Program Hafızası

Şekil 3.2'de program hafızasının ilk kısmı görülüyor. Bir resetten sonra, CPU işleme 0000H adresinden başlar. Aynı şekilde görüldüğü gibi her işkesme (interrupt) program hafızasında sabit bir yere atanmıştır. Bir işkesme geldiğinde CPU bu bölgeye atlar ve işkesme servis yordamını buradan başlatır. Eğer işkesme kullanılmayacaksa, bu bölge genel amaçlı program hafızası olarak kullanılabilir. [1]

İşkesme servis bölgeleri 8 byte aralıklarla yerleştirilmiştir. 0003H de harici işkesme 0, 000BH de zamanlayıcı 0, 0013H de harici işkesme 1, 001BH de zamanlayıcı 1 gibi örnekleri vermek mümkündür. Eğer işkesme yordamı yeterince kısa olursa bu bölgeye yerleştirilebilir. (Kontrol uygulamalarında olduğu gibi) Daha uzun servis yordamlarında, eğer diğer işkesmeler de kullanılıyorsa, sonraki işkesme bölgesini işgal etmemek için "JUMP" komutu kullanılmalıdır.

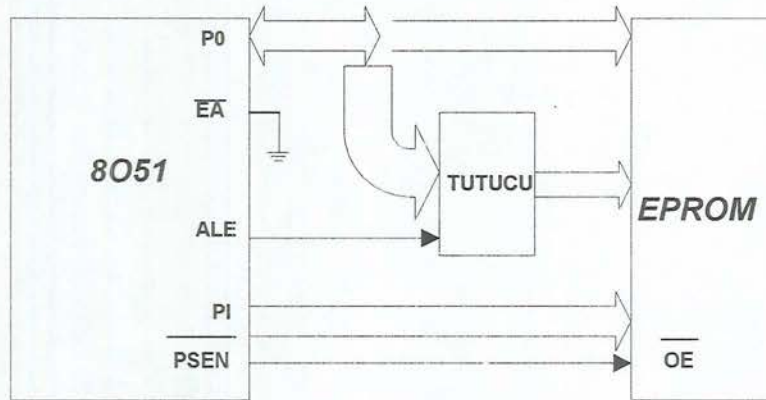
Program hafızasının ilk 4Kbyte'lık kısmı çip üzerinde veya harici olarak kullanılabilir. Bu seçim  $\overline{EA}$  (External Access / Harici Erişim) bacağına Vcc veya Vss gerilimleri uygulanarak yapılır.  $\overline{EA}$  bacağına Vcc gerilimi uygulanırsa, program dahili hafızasının 0000H'den 0FFFH'e kadar olan adreslerini kullanır. Programın kullandığı harici hafıza, 1000H ile FFFFH adresleri arasında kalır.



Şekil 3.2 : Program hafızası

$\overline{EA}$  bacağına  $V_{ss}$  gerilim uygulanırsa, tüm program direkt olarak harici hafızayı kullanacaktır. ROM bulunmayan tiplerde (8031, 80C31 gibi) harici hafızanın kullanılması için bu bacak  $V_{ss}$  gerilimine çekilmelidir.

Harici ROM'a gönderilen okuma sinyali  $\overline{PSEN}$  tüm harici program işlemleri için kullanılır.  $\overline{PSEN}$  dahili program işlemleri için aktive edilmez.



Şekil 3.3 : Harici program hafıza bağlantısı

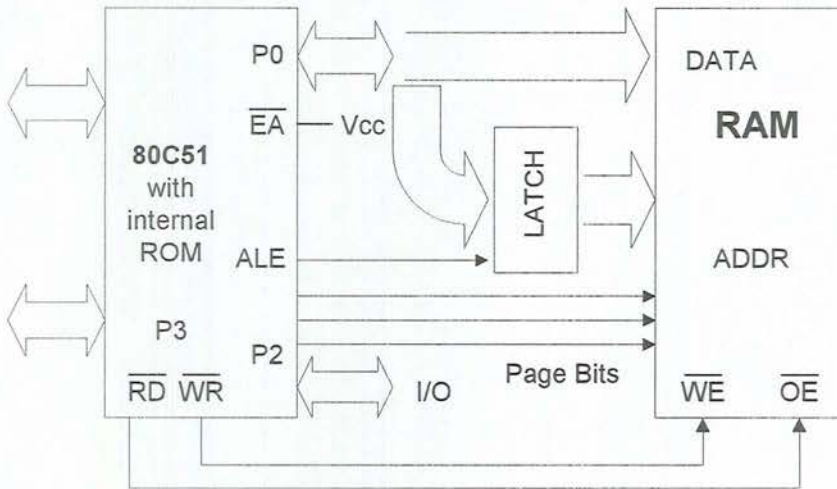
Harici program işletimi için kullanılan donanım yapısı Şekil 3.3'de görülmektedir. Görülüyor ki, program işletimi sırasında 16 I/O biti (P0 ve P2) veriyolu işlevini görüyor. Port 0 (P0) çoklanmış adres/veri yolu olarak kullanılıyor. P0, program sayıcının (PCL) ilk byte'ını adres olarak gönderir ve program hafızasından gönderilen kod kendisine ulaşınca kadar bekler. Program sayıcının ilk byte'ı P0 da hazır iken ALE (Adres Latch

Enable) sinyali bu bilgiyi bir adres latch'inde kilitler. Aynı zamanda P2 ise program sayıcının son byte'ını (PCH) gönderir.  $\overline{PSEN}$  sinyali EPROM'u uyarır ve kod byte'ı okunur.

Program hafızası adresi, toplam hafıza 64Kbyte'ın altında iken bile daima 16 bittir. Harici program kullanımında 8 bitlik portlardan iki tanesi (P0 ve P2) program hafızasını adreslemek için kullanılır.

### 3.1.5. Veri Hafızası

Veri hafızasının adresi DPTR registerinde bulunmaktadır. 64Kbyte'a kadar harici hafıza elemanı kullanılabilir. Veri hafızasının en küçük 128 byte'lık bölümü 80C51'in üzerinde bulunmaktadır.  $\overline{RD}$  ve  $\overline{WR}$  bacakları ile veri hafızasının okuma ve yazma işlemleri yapılır. Şekil 3.4'de şematik gösterim verilmiştir. [1]

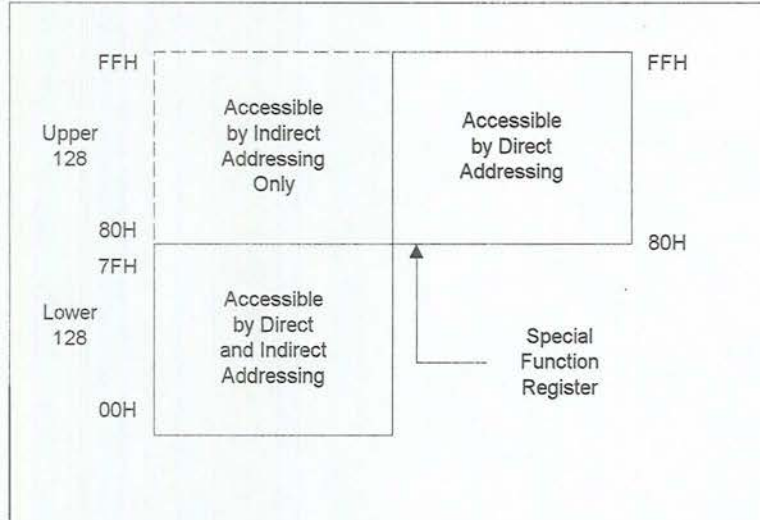


Şekil 3.4 : Harici veri hafıza bağlantısı

Dahili veri hafızasının haritası Şekil 3.5'de verilmiştir. Bu hafıza üç ayrı kısma ayrılmıştır. Bunlar alt 128 byte, üst 128 byte ve SFR (Special Function Registers)'dir.

Dahili veri haritası tek bir byte ile adreslenir. 7FH adresinden yukarı olan SFR hafıza bölümünde direkt adresleme, diğer 7FH'den yukarı olan bölümde indirekt adresleme, 7FH'den düşük olan bölümde ise direkt ve indirekt adresleme yapılır. İndirekt adresleme, seçilen register bankasının R0 ve R1 registerlerinin içine yazılan adreslerle ulaşılır.

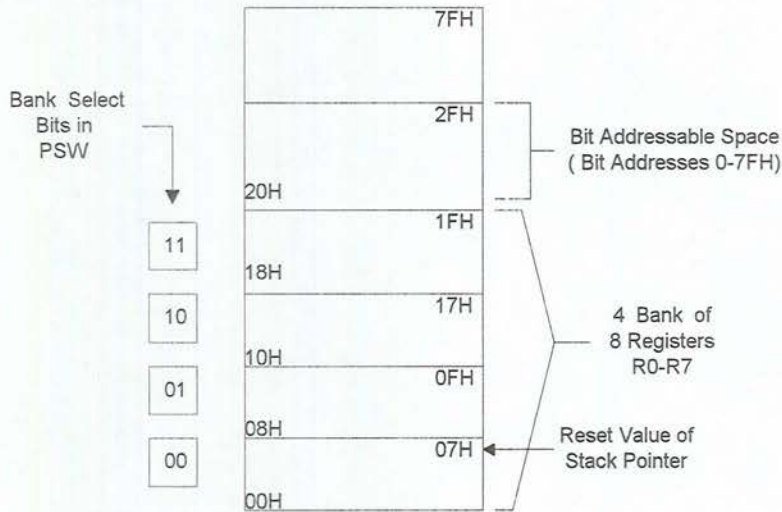
7FH'den yukarı olan bölümler aynı adres boşluklarında olsalar da fiziksel olarak birbirlerinden ayrılmışlardır.



Şekil 3.5 : Dahili veri hafızası

7FH'den aşağı olan bölümün haritası Şekil 3.6'da verilmiştir. Burada 4 bankaya ait 8'er register bulunmaktadır. Toplam 32 register vardır. Bu registerler komut setinde R0,R1,...,R7 olarak isimlendirilir. Register bankasının seçimi PSW'nin (Program Status Word) içindeki ilgili bitler tarafından yapılır. Direkt adresleme komut uzunluğu olarak daha kısa olduğundan bu adresleme kullanılır.

1FH'den sonra gelen 16 byte, bitleri adreslenebilir bytelardan oluşur. Bu bitleri kullanabilmek için 80C51 komut setinde tek bitlik işlemlerle ilgili komutlar bulunmaktadır. Bu bitlerin adresleri bu alana 00H'den 7FH'e kadardır.



Şekil 3.6 : Dahili RAM'in alt 128 byte'ı

SFR'lerin bulunduğu veri hafızasındaki registerlerden başlıcaları port tutucuları, zamanlayıcı içerikleri, durum ve kontrol kelimeleri, arayüz kontrol kelimeleridir. Bu registerlerden 16'sı direkt adresleme ve bit adresleme metoduyla kullanılabilir. Bunların adreslerinin son haneleri X0H ve X8H ile biter.

### 3.1.6. 80C51 İle Seri Haberleşme

Alınan ve gönderilen verinin ikisinde aynı adrestedir, fakat alıcı ve verici registerleri fiziksel olarak ayrılmıştır. Eğer SBUF'a bir bilgi yazılıyorsa bu verici registerine yazılır, SBUF'tan bir bilgi okunuyorsa alıcı registerinden okunur. [1]

80C51 ile seri haberleşmenin dört durumu vardır:

**MOD 0** : Seri bilgi girişi ve çıkışı RxD bacağından yapılır. TxD bacağı kaydırıcı saat çıkışıdır. Standart olarak 8 bit alıcı/verici yapılıdır. Burada LSB bitinden gönderilmeye ve alınmaya başlanır. Baud oranı osilatör frekansının 1/12'sine eşittir.

**MOD 1** : TxD pini ile 10 bit gönderilir veya RxD pini ile alınır. Bir tane başlama biti (0), 8 adet veri biti (önce LSB) ve durma biti (1) formatındadır. Durma biti bilgi alma işlemi yapılırken SCON'un içindeki RB8 bitine yazılır. Baud hızı değişebilir.

**MOD 2** : TxD pini ile 11 bit gönderilir veya RxD pini ile alınır. Bir tane başlama biti (0), 8 adet veri biti (önce LSB) programlanabilir. 9. bit ve durma biti (1) formatındadır. Verici durumunda 9. bit 0 veya 1 olabilir, bu bit SCON'un içindeki TB8'de yer almaktadır. Alım sırasında 9. bit SCON'nun içindeki RB8 bitine gelir. Baud oranı osilatör frekansının 1/32'sine veya 1/64'üne programlanır. Bu PCON'un içindeki SMOD bitine bağlıdır; 0 ise 1/64, 1 ise 1/32'dir.

**MOD 3** : TxD pini ile 11 bit gönderilir veya RxD pini ile alınır. Bir tane başlama biti (0), 8 adet veri biti (önce LSB) programlanabilir. 9. veri biti ve durma biti (1) formatındadır. Mod 3, aynen mode 2 gibidir, tek farkı mod 3'ün baud oranının değişebilir olmasıdır.

Bütün bu modlarda gönderme bilgisi SBUF'a yazılır. Alıcıda ise mod 0 için R1=0 ve REN=1 olması gerekir. Diğer modlardaysa REN=1 ise başlama biti gelmeye başlar. 80C51'le çoklu haberleşmede mod 2 ve mod 3 kullanılır.

**SCON Registeri (Serial Port Control Register) (98H)**: Seri port kontrolünün yapıldığı ve durumunun okunduğu registerdir. Bu register durum seçim bitlerini, 9. veri bitlerini (RB8, TB8) ve seri port işkesme bitlerini (TI, RI) içerir.

**SCON :**

MSB				LSB			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

**SM2** : Mod 2 ve 3 için çoklu haberleşme enable biti, 0 ise disable, 1 ise enable'dir.

**REN** : Seri alma enable biti. Programla yazılır ve silinir.

**TB8** : Mod 2 ve 3'deki 9. veri bitidir. Programla yazılır ve silinir.

**RB8** : Mod 2 ve 3'deki alınan 9. veri bitidir. Mod 1'de eğer SM2=0 ise, alınan durma biti buraya yazılır. Mod 0'da kullanılmaz.

**TI** : Donanım tarafından set edilir. Mod 0'da gönderilen 8 bitten sonra set edilir. Diğer modlarda durma biti gönderildiğinde set edilir. Programla sıfırlanması gerekir.

Çizelge 3.2 : SCON Register bitlerinin tanımlanması

SM0	SM1	Mod	Tanımlama	Baud Oranı
0	0	0	Kayıt Registeri	$f_{osc}/12$
0	1	1	8-bit UART	Değişken
1	0	2	9-bit UART	$F_{osc}/64$ veya $f_{osc}/32$
1	1	3	9-bit UART	Değişken

### 3.1.7. 8051'in EPROM Ürünleri

Tüm EPROM ürünleri pencereli DIP (Dual In-line Package) veya OTP (One Time Programmable) olarak bulunabilir. Pencereli DIP konfigürasyonu, EPROM'un güçlü bir morötesi ışık kaynağı altında silinmesine olanak tanıyarak program geliştirmeyi hızlı ve kolay kılar. OTP sürümü silinemez, çünkü morötesi ışınları alacağı bir penceresi yoktur. OTP konfigürasyonu bir kereye mahsus programlanabildiğinden pencereli DIP'lere göre daha ucuzdur. Bu nedenle maskeli ROM türlerinin arzulanmadığı durumlarda avantaj sağlar.

## 3.2. DC Servo Motor

### 3.2.1. Servo Sistem Nedir

Servo kavramı; Yunanca Servus'tan gelir, ve servo sistem diye adlandırılır. Bu sistem verilen komutlara güvenilir bir şekilde cevap verir. Kısaca; mekanik sistemlerin belirlenmiş komut değerleri ile hedef pozisyonları ve hız değişimlerini uygun bir şekilde kontrol eden sistemdir. [4]

### 3.2.2. Servo Sistem İle Kontrol

Servo sistemler mekanik sistemlerde hız ve pozisyon kontrolü için tek alternatif değildir, fakat diğer sistemlere nazaran pozisyon ve hız kontrolünde en iyi çözüm sunan sistemlerdir. [4]

Hemen hemen bütün mekanik sistemler, ileri ve geri dönüşlerde tarafsız bir bölgeye sahiptir. Örneğin bir sistemi ileri dönüş yönünden, geri dönüş yönüne çevirmek istediğimizde, ileri ve geri dönüş arasında, yön değişimine bağlı olarak bir bekleme zamanı olur. Sistem ani dönüşü izin vermez. Fakat servo sistemlerde ani olarak ters yöne dönüş gerçekleşir. Bu özellik servo sistemleri diğer mekanik sistemlerden üstün tutar.

Son yıllarda bilgisayar kontrollü makine kontrolü Türkiye endüstrisinde yaygın olarak kullanılmaya başlanmıştır. Bilgisayar nümerik kontrol (CNC) teknolojisi; takım tezgahlarında, robotlarda, konveyörlarda ve her türlü hız ve pozisyon kontrolü yapan makinalarda yaygın olarak kullanılmaktadır. Teknoloji ilk defa 1954 yılında bir Amerikan Üniversitesinde (MIT) tamamen elektronik devreler kullanılarak geliştirilmiştir.

1970'lerdeki mikrobilgisayar teknolojisindeki atılımlara paralel olarak katı elektronik devrelerin işlevini birden fazla mikroişlemci yüklenip, günümüzdeki bilgisayarlı nümerik kontrol sistemlerine geçilmiştir. CNC üniteleri programlanabilen mantık kontrol (PLC) devreleri ile entegre edilerek çok eksenli sistemlerde her türlü işlevi yerine getir hale gelmiştir. [5]

Bir sistem üç ana üniteden oluşur: mekanik aksam, güç besleyici amplifikatör ve motorlar, bilgisayar üniteleri. Her ana mil sisteme bir hareket verdiğinden, sistemin serbestlik derecesi eksen sayısı ile belirtilir. Her ana mile bir servo motoru (doğru veya alternatif akımlı) direkt veya bir kademeli redüktör aracılığıyla bağlanır. Doğru akımlı servo motorlar hız kontrolunun kolaylığından dolayı tercih edilmekle birlikte, son zamanlarda alternatif akımlı servo motorların da hassas hız kontrolü mümkün olmaya başlamıştır. Sistemimizde doğru akım motorları ile çalışılmıştır. Doğru akım motorları sistemin hız ve moment gereksinimine göre seçilir.

Servo motoru hızlandırma ve yavaşlatma sırasındaki dinamik ve statik yükleri istenilen hızlarda taşıyabilmelidir. Hızlanma sırasında sisteminin ataletini istenen ilerleme hızına getirirken, gelecek dinamik moment ( $J_e \cdot d\omega/dt$ ), statik sürtünmeden gelen sabit moment ( $T_f$ ) ve viskoz sürtünmeden ( $B\omega$ ) gelen hızla orantılı değişken momentlerin toplamı motora yük olarak gelir. Seçilen motorun moment kapasitesi toplam yük momentlerinin üstünde olmalıdır. Motora en fazla yük ivmelendirme sırasında ortaya çıkan, atalet momentiyle orantılı olan dinamik momentlerdir. Motorun güç besleyici devresinin yaklaşık 2 saniye zaman aralığı süresince normalin üstünde moment verebilme kapasitesi vardır ve ivmelendirme sırasında kullanılır. Sistem sabit bir hıza ulaştınca dinamik moment sıfır olur ve sadece sürtünmeden gelen statik-sabit momentler motoru yükler ki bunlarında şiddeti düşüktür. Motorun sürekli moment kapasitesi bu statik yükleri taşıyabilecek şekilde seçilmelidir.

### 3.2.3 Servo Sürücü Devresi

Motorun dönme hızı ( $\omega$ ), kutuplarına verilen doğru akımlı voltajla orantılıdır. Motorun ürettiği moment sabiti ( $K_t$ ) motor üreticisi tarafından belirtilir. Motora gelen doğru akımlı voltaj güç besleyici devresinden sağlanır. Servo sürücü devresi, akım güç devresi, doğru akım voltaj güç devresi ve PWM (vuru aralıklı modülatör) devrelerinden oluşur. Servo motorlara takometre ve enkoder bağlanabilir. Takometre motorun dönme hızı ile orantılı olarak belli bir voltaj üretir, bu voltaj ( $x_3$ ) servo sürücü devresinde sistem bilgisayarından gelen komut hız voltajı ile karşılaştırılır. Fark, diğer adıyla hız hatası ( $e_1$ ), akım güç devresinden geçirilir. Genelde servo sürücü devresinde armatürden ölçülen motor akımı ( $i$ ), güç devresinden üretilen akım ( $x_1$ ) ile karşılaştırılır. [6]

Akımın geri beslemesi  $K_a$ , motorun statik momentleri sürekli olarak yenebilmesi ve hızda düşüş olmaması içindir. Akım hatası, voltaj ve PWM güç devresinden ( $K_v$ ) geçirilir. PWM devresinin verdiği dalgalar 8000Hz'nin üstünde olmalıdır, yoksa modülatörün çıkardığı ses gürültüsü operatörü rahatsız edici şiddette olur. Akım ve voltaj güç devrelerinin verebileceği değerler, kapasitelerine göre sınırlıdır. İstenen fazla akım ve voltaj yerine limit değerler salgılanır. PWM'den salgılanan dalgalı voltaj ana sisteme bağlı doğru akım motorunun kutuplarına kablo ile iletilir. Dalgalı voltajın ortalama şiddeti, motorun sarım parametreleri olan, empedans ( $L_a$ ) ve direnç ( $R_a$ ) bağımlı olarak

armatürde akım ( $i$ ) oluşturur. Motorun verdiği moment ( $T_m$ ), armatürdeki akım şiddeti ( $i$ ) ve motor sabiti ( $K_t$ ) ile orantılıdır. Motor momentinin bir kısmı hareket yönü ( $\text{sgn}(\omega)$ ) ile işareti değişen sabit Coulomb sürtünmesinin ( $T_f$ ) motor miline iletilen momentini ( $T_c$ ) yenmek için harcanır. Geri kalan moment sistem motor mili tertibatının ataletini ( $J_e$ ) ve viskoz sürtünmeyi ( $B$ ) yenmek için kullanılır. Sonuçta motor ve ona bağlı olan ana mil dönmeye başlar. Bu dönme hızına ( $\omega$ ) orantılı olarak ( $K_b$ ) motorda elektro-motor voltajı oluşur.

Motorun dönme hızının kontrolü demek sistem ilerleme hızının kontrolü demektir ki bu da motora verilen ortalama doğru akım voltajının sürekli otomatik kontrolünü gerektirir. Servo sürücü devresinin görevi budur.

Sistem ilerledikçe PWM devresindeki impulslar devamlı değişerek sistemin gerçek pozisyonunu, enkoderler sayesinde iki yönlü dijital sayıcıda kaydedilir. Bu işlem, otomatik kontrol devresinde integrasyon işlemi ( $1/s$ ) olarak modellenir. Sistem bilgisayarı yarı dijital enterpolasyon alt programından temel uzunluk birimi olarak belirli zaman aralıklarıyla gelen pozisyon kumandalarını enkoder sayıcısındaki değerle karşılaştırır. Pozisyon komutundaki ani vuruların zaman aralığı o eksenin ilerleme hızına bağlı olarak bilgisayardaki enterpolasyon altprogramı tarafından belirlenir. Salgılanan pozisyon komutu ile enkoder sayıcısındaki gerçek pozisyon değeri sabit dijital servo kontrol zamanı ( $T$ ) aralıklarıyla okunarak pozisyon hatası hesaplanır. ( $e_1$ ) Pozisyon hatası dijital servo algoritmasından geçirildikten sonra dijital/analog (D/A) devresinden geçirilerek voltaj karşılığı servo sürücü devresinin hız komutu olarak kablo ile verilir.

Bu şekilde sistemin bir ilerleme milinin hız ve pozisyonunu sağlayan servo sürücü devresi tamamlanmış olur. Sistem durdurulacağı zaman, sistemin ataleti dikkate alınarak 0.1 saniye kadar önce hız doğrusal olarak yavaşlatılmaya, yani enterpolasyon algoritması daha aralıklı zamanlarla puls üretmeye başlar. Sonunda toplam pozisyon adedi kadar puls üretilince, dijital enterpolasyon komut üretmeyi durdurur ve sistemin enkoder sayıcısı da sistemde komut kadar ilerlediğini belirtirse pozisyon hatası sıfır olur. Bu sayısal sıfır hata değeri, D/A devresinin servo sürücü devresi için sıfır voltaj üretilmesine sebep olur. Sonunda motorun kutuplarına gelen voltaj sıfır olur ve ilerleme yumuşak bir şekilde istenen pozisyonda durulur.

### 3.3. Enkoder

Motorun miline bağlı olan enkoder, üzerinde yarıklar olan bir disk ve optik okuyucudan oluşmuştur. Enkoderde birbirine göre hafifçe kaydırılmış (faz farkı olan) iki adet disk vardır. Bu açı farkı sistemin ileri veya geri (saat yönü veya saatin ters yönü) hareketlerini hissetmek için yapılır. Örneğin bizim sistem enkoderindeki iki diskin her birinde 512 adet yarık vardır, dolayısıyla toplam yarık miktarı 1024 adettir. Mil döndükçe her yarık dikdörtgen bir vuru (pulse) verir. Enkoderin verdiği bu vurular motorun göbeğinden alınıp kablo ile sistem bilgisayarına aktarılır. Bilgisayardaki enkoder/dekoder entegre devreleri dönüş yönünü saptayıp her vurudan iki adet ani bir vuru (impulse) sinyali üretir. Dolayısıyla motor milinin bir dönüşünde enkoder-dekoder devresi 2048 adet ani bir vuru sinyali üretir. [3]

## 4. SİSTEM YAZILIMI

### 4.1 Giriş

Sistem donanımını idare etmek, verileri kontrol etmek, karşılaştırmak, hesaplamaları yapmak, gerekli bilgileri hafızaya yazmak ve okumak, sistemdeki arızaları tespit etmek, sisteme ait kontrolleri düzenlemek ve uygulamak gibi işlemleri gerçekleştiren, mikrobilgisayar kartında merkezi işlem birimi (CPU) olarak kullanılan, Intel 80C51AH mikroişlemcisinin en büyük özelliği, yüksek seviyeli bir dil olan Basic51 (BXC51) ile programlanabilmesidir. Bu özellik, program geliştirmede büyük kolaylıklar sağlamaktadır. [8]

Sistem için program Ek-1 ve Ek-2'de görüldüğü gibi yazılmıştır. Bu tür program yazımı BXC51'in özelliğinden kaynaklanmaktadır. Program, temel programlama dili olan "Basic" ile aynıdır. Programa, program numarası verilerek yazılmaya başlanır. Program yazılımı sırasında, basic dili yerine assembler diliyle programa devam etmek, program yazılımını kolaylaştıracak veya basitleştirecek ise; programda kalınan satırın altına: "\$ASM" yazılarak, assembler diliyle programı yazmaya devam edilir. İstenildiği kadar assembler diliyle program yazımına devam etmek mümkündür. Ne zaman ki basic diline ihtiyaç duyulursa, o zamanda programda kaldığımız yerin altına: "\$BASIC" yazarak tekrar basic dili ile program yazmaya devam edilir. İsteğe bağlı olarak, yukarıdaki işleme benzer şekilde, tekrar program yazma dilini değiştirebiliriz.

Program yazılırken, yazılan kısımların denenmesi istenirse; yazılmış olan kısım; "SIM51" adlı program altında, aşağıda verilen program komutu ile assembler koda çevrilir.

DERLE <Program Adı>

Daha sonra derlenmiş olan bu program, SIM51 simülör programı içerisinde "compile" edilir. İstenilen şekle göre adım adım veya programın tamamı çalıştırılabilir. SIM51 simülör programı içinde mikroişlemciye ait registerlerin, zamanlayıcıların, sayıcıların, portların,..vb. kısımların içeriklerini görmek mümkündür.

Eğer tasarlanan sisteme ait donanım mevcut ise; bu durumda PROCOM PLUS programı ile sistemi denemek mümkündür. Bunun için; kullanılan bilgisayarın seri portundan, RS232 arabirimi ile emülatöre bağlantı sağlanır. Yazılan program, procom plus programı altındaki aşağıda verilen program komutu ile assembler koda çevrilir.

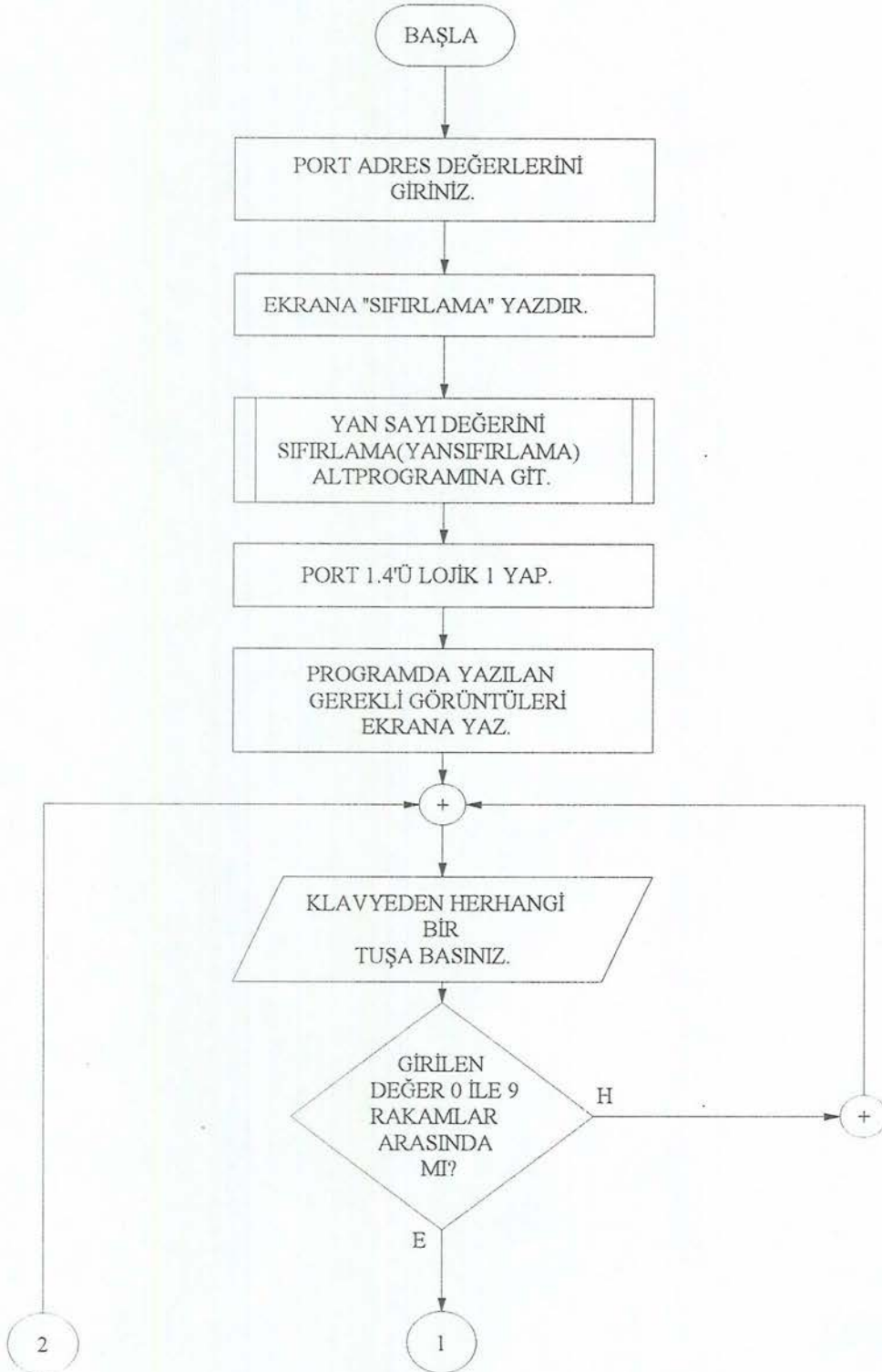
DERLEM <Program Adı>

Bilgisayarda assembler koda çevrildikten sonra klavyeden "L" tuşuna basılır. Sonra klavyeden "Page Up" ve "A" tuşuna basılır. En son olarak, programı çalıştırmak için ekrana G 2000 yazmak yeterlidir. Sistemle ilgili değerleri ekranda görmek mümkündür.

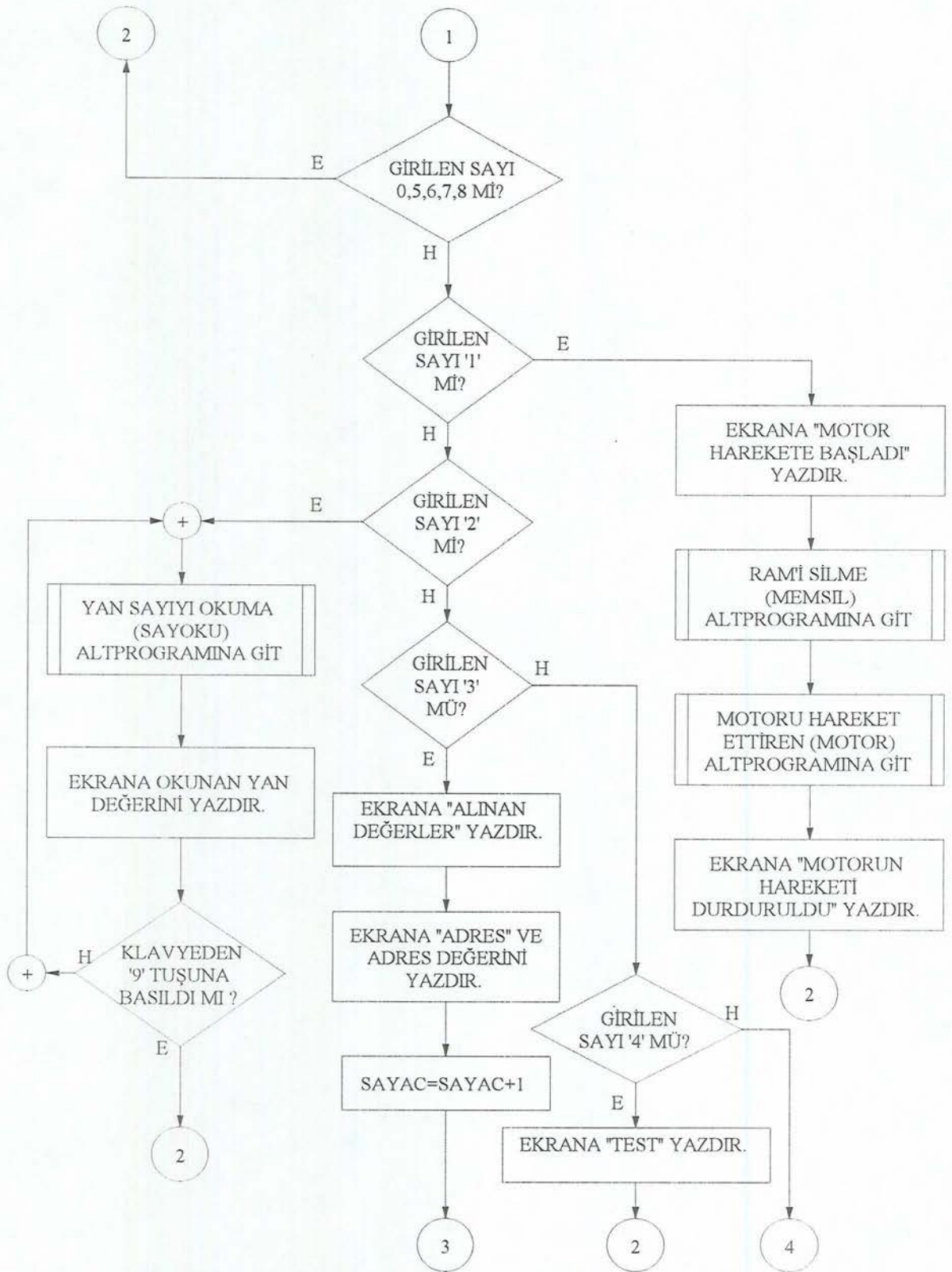
Sistemde kullanılacak olan servo motorun hareketini bulma programı Ek-1'de verilmiş ve bu programın akış diyagramı Şekil 4.1'de ve programın açıklanması Bölüm 4.2'de yapılmıştır.

Sistem kontrol programı adı altında, bir program yazılmıştır. Yazılmış olan bu program Ek-2'de verilmiştir. Bu programın akış diyagramı Şekil 4.5'de ve programın açıklanması Bölüm 4.3'de yapılmıştır.

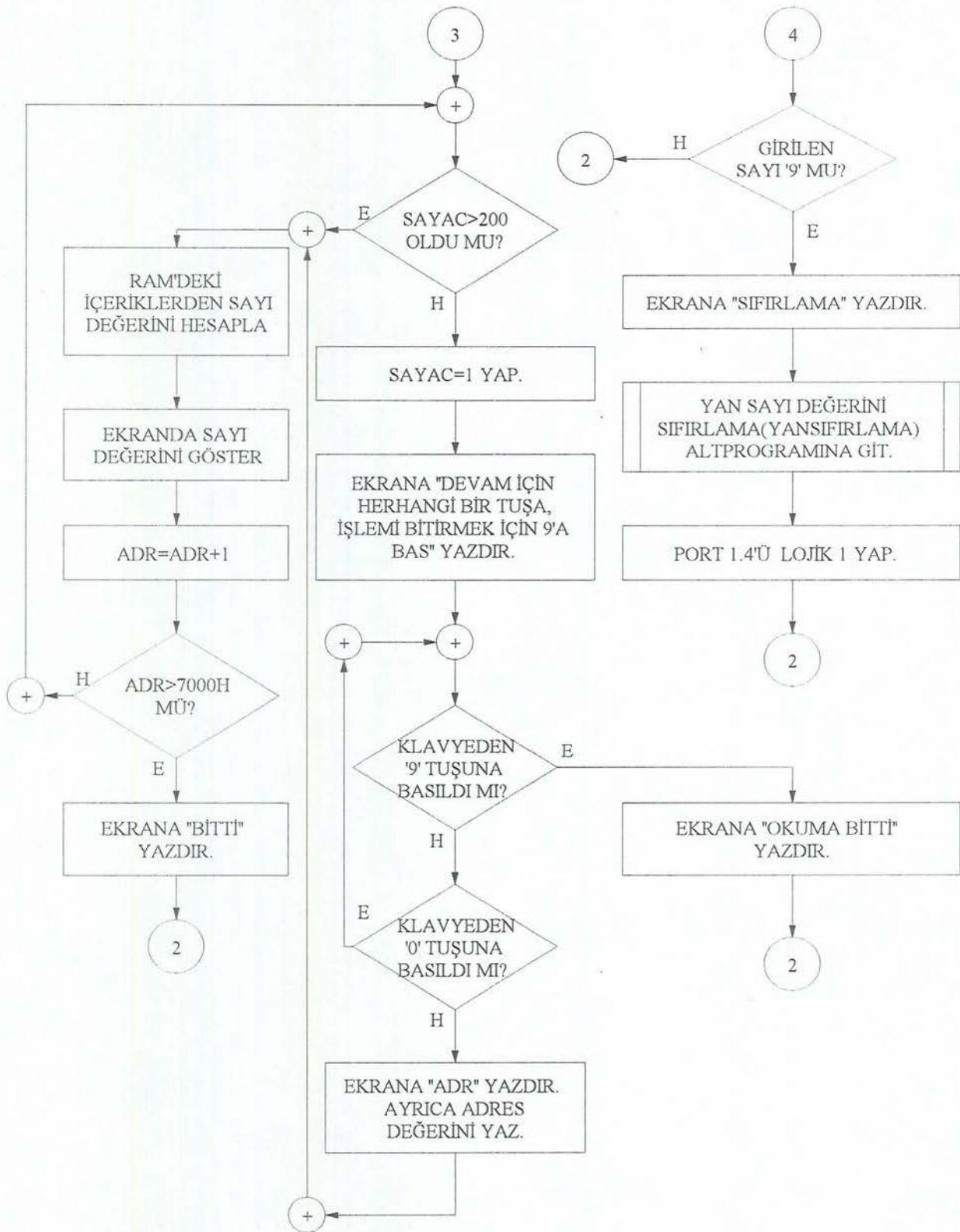
#### 4.2. Motorun Transfer Fonksiyonunu Bulma Programı



Şekil 4.1 : Dc servo motorun transfer fonksiyonunu bulma programı akış diyagramı



Şekil 4.1 : Transfer fonksiyonunu bulma programı akış diyagramının devamı



Şekil 4.1 : Transfer fonksiyonunu bulma programı akış diyagramının devamı

İlk olarak programda kullanılacak olan Port A, Port B ve Port C'nin adresleri belirlenir. Bu adres belirlemesini dekoder (74 HCT 138) çıkışlarına bakarak yapılır. Adresleme bitlerinden A13, A14, A15 dekoderin girişine bağlanır. Dekoderde sekiz ayrı çıkış mevcuttur. Bu çıkışlar da devredeki Port A'ya bağlı D tipi flip-flop (U7-74 HCT 573) entegresine, Port B'ye bağlı D tipi flip-flop (U8-74 HCT 573) entegresine, Port C'ye bağlı D tipi flip-flop (U9-74 HCT 573) entegresine ve RAM'e (U4-6264) bağlıdır. (Bakınız Şekil 4.2) Sisteme ait adresleme şeması aşağıda tablo halinde verilmiştir. [2]

Çizelge 4.1 : CPU kartı dekoder adresleme yapısı

Dekoder Girişi			Dekoder Çıkışı	Adresleme Aralığı	Adreslenen Entegreler
A2	A1	A0			
0	0	0	Q0	0000H-1FFFFH	RAM - 6264 (U4)
0	0	1	Q1	2000H-3FFFFH	
0	1	0	Q2	4000H-5FFFFH	
0	1	1	Q3	6000H-7FFFFH	
1	0	0	Q4	8000H-9FFFFH	U9 - OUC1
1	0	1	Q5	A000H-BFFFFH	U7 - INA
1	1	0	Q6	C000H-DFFFFH	U8 - INB
1	1	1	Q7	E000H-FFFFFFH	

Sistemin sağlıklı işleyebilmesi için ilk olarak programın başında sistemi sıfırlama (SIFIRLAMA bölümü) adlı altprograma dallanma sağlanır. Programda sadece yan motora ait denemeler yapılacağı için, yan motor enkoder değerlerini sıfırlayan YANSIFIRLA adlı altprogram kullanılır. Bu altprogramda Port 1'e 10H (0001 0000B) sayısı girilir. (Açıklama 1) Enkoder sayım kartında Port 1'e bağlı U3 dekoder entegresi bulunmaktadır. Port 1.0, Port 1.1 ve Port 1.2'ye lojik 0 seviyesi verilirse, dekoderin D0 çıkışı seçilmiş olur. D0 çıkışı U1 ve U2 entegrelerinin OC girişine bağlıdır. OC sinyal girişlerine gelen seviye lojik 0'dır. Dolayısıyla U1 ve U2 entegreleri çıkış vermezler. Port 1.4'e, motor sürücü devresindeki BC 327 tipi transistörün baz bacağı bağlıdır. (Bakınız Şekil 4.3) Transistörün bazına lojik 1 seviyesi gelmesi, transistörün iletme geçmesine sebep olur. Bu durumda MOC3020 opto-kuplör entegresinin 4 nolu bacağı aktif olur. Buna bağlı olan BD135 transistörünün bazı aktif olur ve transistör iletme geçer. Buna bağlı olan rölenin bobini enerjilenir. Motorun (+) ucu rölenin NC (normali kapalı) ucuna bağlıdır. Dolayısıyla röle bobini enerjilendiği zaman, motora +24V'luk gerilimin gitmesi engellenmiş olur. Böylece enkoder sayım kartının ve motorun devreden çıkması sağlanmış olur.

Daha sonra Port 1'e 12H (0001 0010B) sayısı girilir. (Açıklama 2) Enkoder sayım kartında Port 1'e bağlı U3 dekoder entegresi bulunmaktadır. (Bakınız Şekil 4.4) Port 1.0 ve Port 1.2'ye lojik 0 seviyesi, Port 1.1'e lojik 1 seviyesi verilirse, dekoderin D2 çıkışı seçilmiş olur. D2 çıkışı JP3 hattı vasıtasıyla UBA-4070 entegresinin 1 nolu bacağına lojik 1 seviyesinde sinyal gönderir. Böylece UBA-4070 entegresinin 3 nolu bacağı aktif olur. Bu 3 nolu bacak 4516 dört bit sayıcı entegrelerinin (U4, U5, U6, U7), sayıcı sıfırlama bacaklarına paralel bağlıdır. Böylece enkoder sayım kartındaki, sayıcı entegreler sıfırlanmış olur.

Bu işlemler yapılırken sistemin işlemcisi, 1.075 ms'lik bekleme döngüsünde işlemlerin tamamlanması için bekletilir. Döngü tamamlandıktan sonra enkoder sayım kartından dışarıya bilgi akışı olmaması için Açıklama 1'deki işlemin aynısı bir kez daha tekrarlanır. Motorun enerjilenmediğinden emin olmak için tekrar Port 1.4'e lojik 1 seviyesi verilir. Ana programa geri dönülür.

Program içinde bazı özel işlemler için, klavyeden girilen değerlere göre programda dallanmaların yapılması sağlanmıştır. Procom Plus adlı yazılım altında program çalıştırıldığında, ekranda hangi işlem için hangi tuşa basılması gerektiği ayrıntılı olarak görülmektedir. Ekranda görünen işlemlerden hangisi isteniyorsa onu tanımlayan tuşa basarak dallanma sağlanır.

Klavyeden sayı girilene kadar program bu noktada bekler. Klavyeden girilen sayının 0 ile 9 arasında bir rakam olup olmadığı kontrol edilir. Bu sayıların haricinde bir tuşa basılmış ise döngüde kalmaya devam eder. (Programda 50, 55, 60 nolu satırlar) Klavyeden istenen sayılar girilmiş ise programda 190 nolu satıra dallanma sağlanır.

Klavyeden girilen sayı değerine göre, programın bu satırlarında çeşitli adreslere dallanmalar meydana gelir. Örneğin; klavyeden 0, 5, 6, 7 ve 8 sayıları girilirse programda 196 nolu satıra dallanma sağlanır.

Klavyeden "1" nolu tuşa basılırsa program 200 nolu satıra dallanır. Ekranda "motor harekete başladı." şeklinde bir mesaj gözlenir. Daha sonraki assembler satırında, alınacak enkoder değerlerinin saklanacağı RAM'in, boş olduğundan emin olmak için hafıza silme (MEMSIL) adı altındaki bir altprograma dallanılır. Bu altprogramda RAM üzerindeki 1000H adresinden, 7000H adresi arasındaki her bir bayta 0000 0000B değeri yerleştirilir. Bu işlem bittikten sonra motorun harekete geçmesini sağlayacak olan MOTOR adı altındaki altprogram işletilir. Bu altprogramda sayım kartı üzerindeki Port 1.4'e, motor sürücü devresindeki BC 327 tipi transistörün baz bacağı bağlıdır. Transistörün bazına lojik 0 seviyesi gelmesi, transistörün iletme geçmesine engel olur. Bu durumda MOC3020 opto-kuplör entegresinin 4 nolu bacağı lojik 0 olur. Buna bağlı olan BD135 transistörünün bazı pasif olur ve transistör iletme geçemez. Buna bağlı olan rölenin bobini enerjisiz kalır. Motorun (+) ucu rölenin NC (normali kapalı) ucuna bağlıdır. Dolayısıyla röle bobini enerjisiz olduğu zaman, motora +24V'luk gerilimin gitmesi sağlanmış olur. Kayıt yapılacak olan enkoder değerlerinin, RAM'in hangi adresine yazılacağı R3 ve R4 registerlerin içeriğinde mevcuttur. Buradan OKU adlı bir altprograma geçilir. (Açıklama 3) Bulunulan program içinden SAYOKU adlı altprograma dallanılır. Bu altprogramda şu işlemler yapılır: P1 portu 1111 0000B sayısı ile lojik AND işlemine sokulur. Bunun nedeni sayım kartı üzerinde bulunan U3 (6273) entegresi, 8 kanal analog çoklayıcıda bulunan G0, G1, G2 ve EN kontrol girişlerinin seçilmesi gerekmektedir. Lojik AND işlemi ile G0, G1 ve G2 kontrol girişleri lojik seviye olarak sıfır değerini alırlar. Buna bağlı olarak D0 çıkışı seçilmiş olur. D0 çıkışı U1 ve U2 (74HCT573) entegresinin OC bacağına bağlıdır. Bu bacak lojik 1 seviyesine çekildiği için U1 ve U2 entegrelerinin bilgi okuma işlemini keser. Bu entegrelerde bilgi okuma işlemi kesildiği sıradaki değerler saklı kalacaktır. Sırasıyla Port B'de okunan değer, binary değer olarak YANO (Yan değer orta baytı) adresine alınır. Daha sonra Port A'da okunan değer, binary değer olarak YANL (yan değer düşük baytı) adresine alınır. Bu işlemler bittikten sonra OKU altprogramına geri dönülür. Bir sonraki enkoder değerini kaydetmek için 10 mikrosaniye bekleme yapılır. Önceden elde edilen YANO ve YANL değerleri sırasıyla

RAM'de istenen adreslere yerleştirilir. Tekrar MOTOR adlı altprograma geri dönülür. DPH değeri Acc'e alınır. Register işkesme (RI) biti set edilmiş ise CIK adlı etikete atlanır. Eğer set edilmemiş ise Acc'deki sayının 4000H olup olmadığı kontrol edilir. Eğer sayı 4000H'den küçük ise M1 etiketine atlayarak aynı işlemler tekrarlanır, ne zaman Acc'deki sayı 4000H olduğu zaman, sayım kartı Port 1.4'e, motor sürücü devresindeki BC 327 tipi transistörün baz bacağı bağlıdır. Transistörün bazına lojik 1 seviyesi gelmesi, transistörün iletme geçmesine sebep olur. Bu durumda MOC3020 opto-kuplör entegresinin 4 nolu bacağı aktif olur. Buna bağlı olan BD135 transistörünün bazı aktif olur ve transistör iletme geçer. Buna bağlı olan rölenin bobini enerjilenir. Motorun (+) ucu rölenin NC (normali kapalı) ucuna bağlıdır. Dolayısıyla röle bobini enerjilendiği zaman, motora +24V'luk gerilimin gitmesi engellenmiş olur. Bu durumda motor yavaşlamaya başlar.

DPTR register çiftine 4000H sayısı yüklenir. Bu sayının düşük baytı R3 registerine, yüksek baytı R4 registerine yüklenir. Daha sonra Açıklama 3'deki işlemler tekrarlanır. Acc'deki sayının 7000H olup olmadığı kontrol edilir. Eğer Acc'deki sayı 7000H'den küçük ise M2 etiketine atlanarak döngüye devam edilir. Eğer 7000H'e eşit ise CIK adlı etikete atlanır. Burada seri tampon (SBUF) değeri Acc'e alınır. Register işkesme temizlenir ve ana programa geri dönülür. Ekranda "Motorun Hareketi Durduruldu" şeklinde bir ifade gözlenir. Program tekrar başa döner.

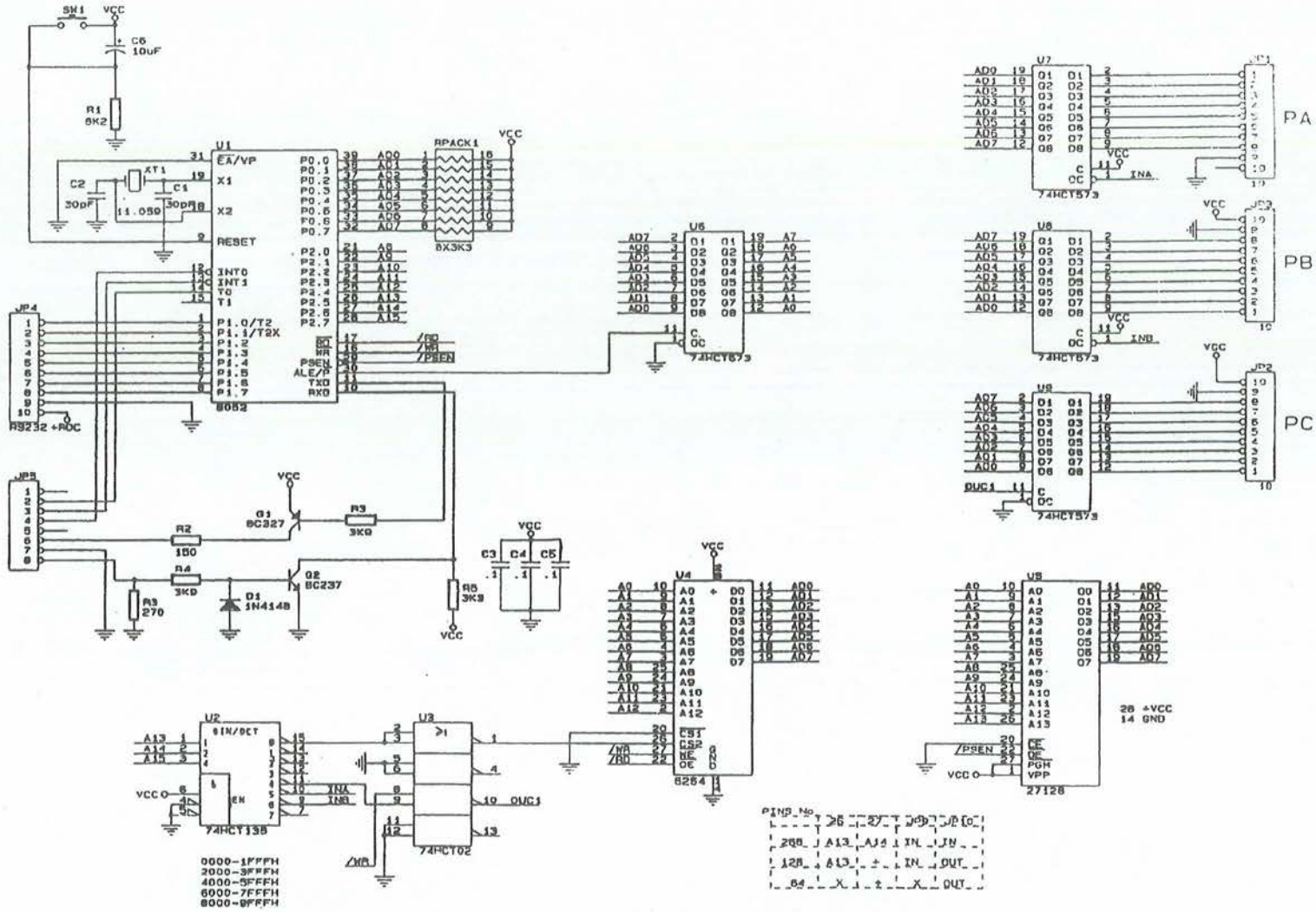
Klavyeden "2" nolu tuşa basılırsa programda 410 nolu satıra dallanma gerçekleşir. Bu bölümde ilk olarak sayıcıyı okuma (SAYOKU) assembler altprogramı işletilir. Bu altprogram Açıklama 3'de ayrıntılı şekilde anlatılmıştır. Bu altprogramdan çıktıktan sonra okunan her enkoder değeri ( $YAN=YANL+256*YANO$ ) ekranda sırasıyla gösterilir. Klavyeden "9" sayısı girilene kadar aynı döngü içerisinde, ekranda değer göstermeye devam eder. "9" sayısı ile birlikte döngüden çıkar ve ana programda başa döner.

Klavyeden "3" nolu tuşa basılırsa programda 300 nolu satıra dallanma gerçekleşir. Bu bölümde RAM'de saklanmış olan değerler ekranda gösterilir. İlk olarak ekranda "Alınan Enkoder Değerleri" şeklinde bir yazı gözlenir. Daha sonra ekranda 1000H adresinden başlayarak 200 adet okunmuş değer gösterilir. Gösterimden sonra ekranda "Devam Etmek İçin Herhangi Bir Tuşa, Durdurmak İçin 9 Tuşuna Basınız" şeklinde bir yazı görünür. Eğer herhangi bir tuşa basılırsa, kaldığı değerden itibaren ekranda göstermeye devam eder. Ne zaman 7000H sayısına ulaşırsa ekranda "Okuma Bitti" şeklinde ifade gözlenir ve ana programda başa dönülür. Eğer "9" nolu tuşa basılırsa ekranda "Okuma Bitti" şeklinde ifade gözlenir ve ana programda başa dönülür.

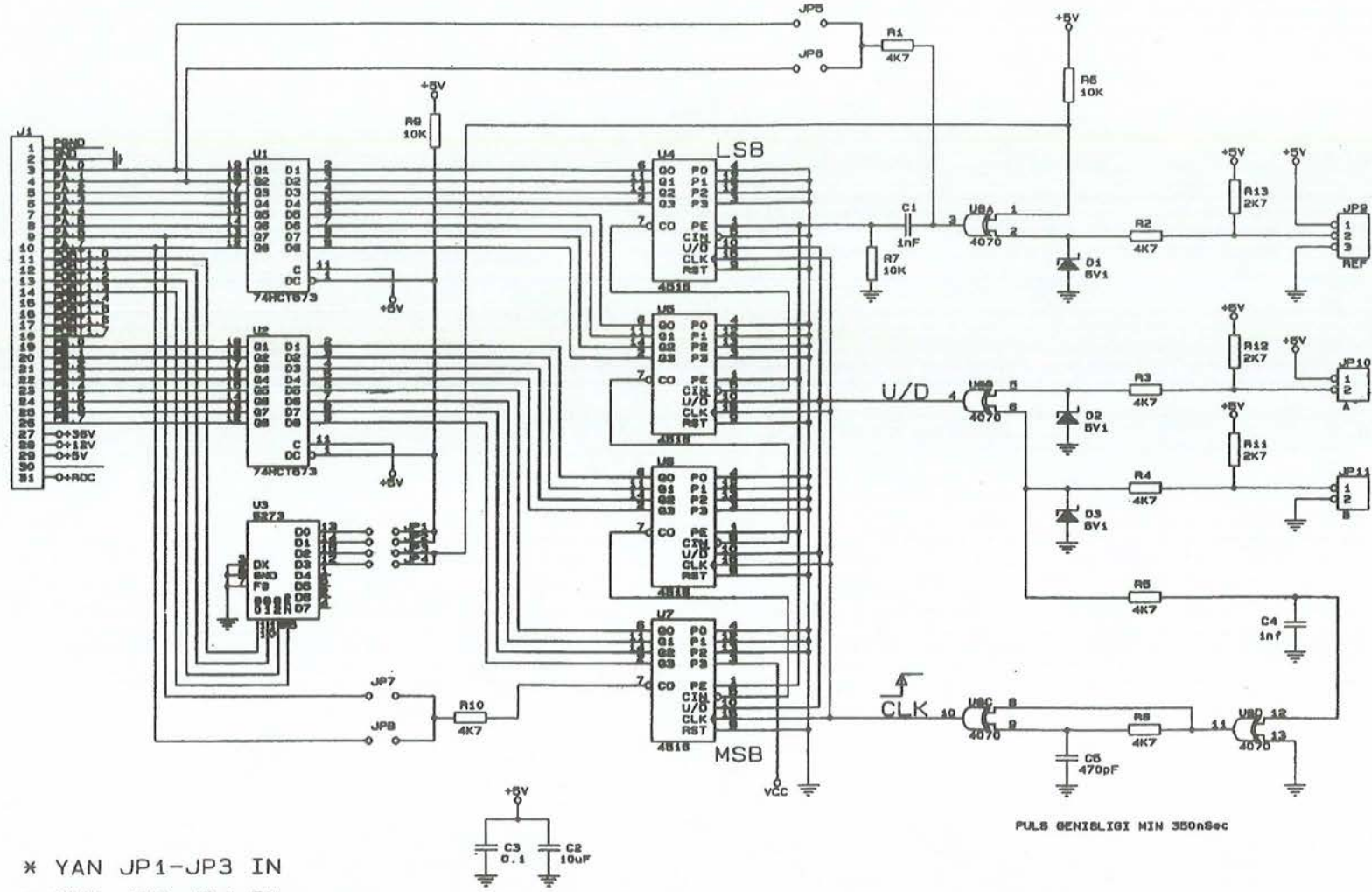
Klavyeden "4" nolu tuşa basılırsa programda 800 nolu satıra dallanma gerçekleşir. İlk olarak ekranda "Sistem Test" şeklinde bir yazı gözlenir ve ana programda başa dönülür.

Klavyeden "9" nolu tuşa basılırsa programda 900 nolu satıra dallanma gerçekleşir. İlk olarak ekranda "Sistem Sıfırlama" şeklinde bir yazı gözlenir. İlk olarak enkoder sayım kartı sıfırlama (YANSIFIRLA) assembler altprogramı işletilir. Bu bölümle ilgili ayrıntılar Açıklama 1'de verilmiştir. Bu altprogramdan sonra ana programda tekrar başa dönülür.

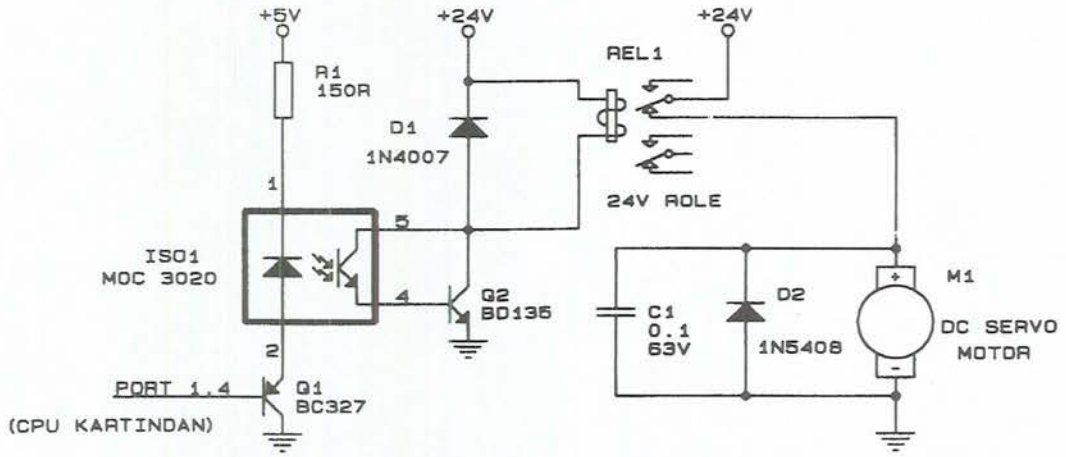
Şekil 4.2: Motor kontrol sistemi CPU kartı devre şeması



Şekil 4.3: Enkoder sayım kartı devre şeması

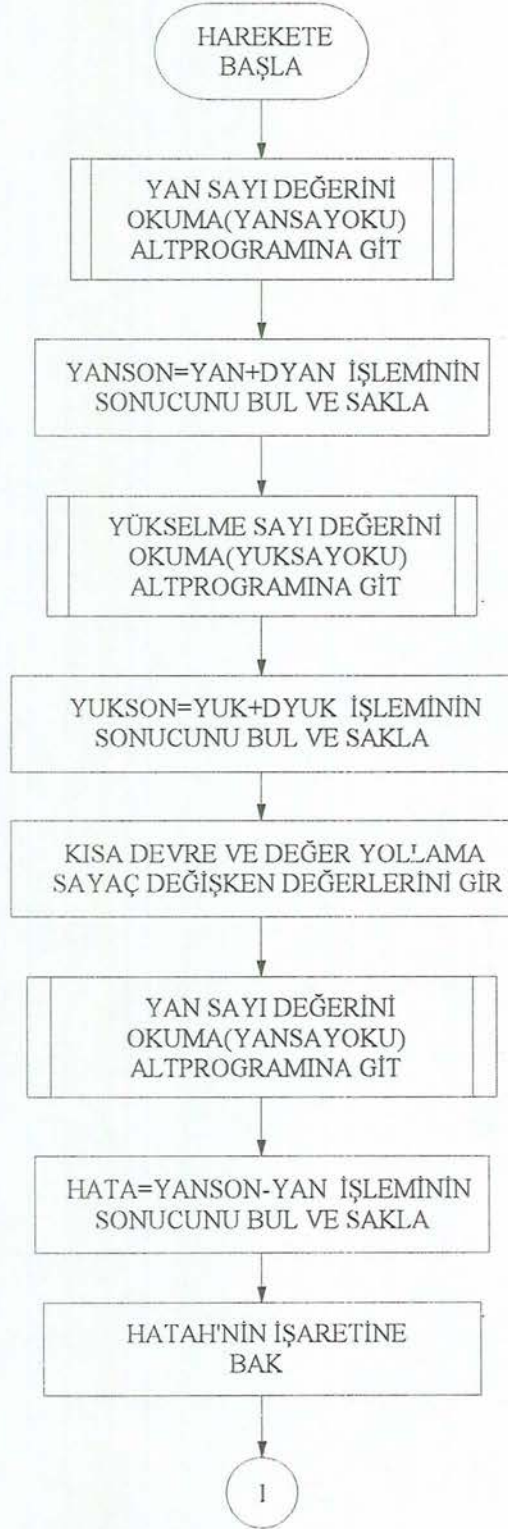


- \* YAN JP1-JP3 IN
- \* YUK JP2-JP4 IN
- \* JP5-JP6-JP7-JP8 TAKILMAYACAK.

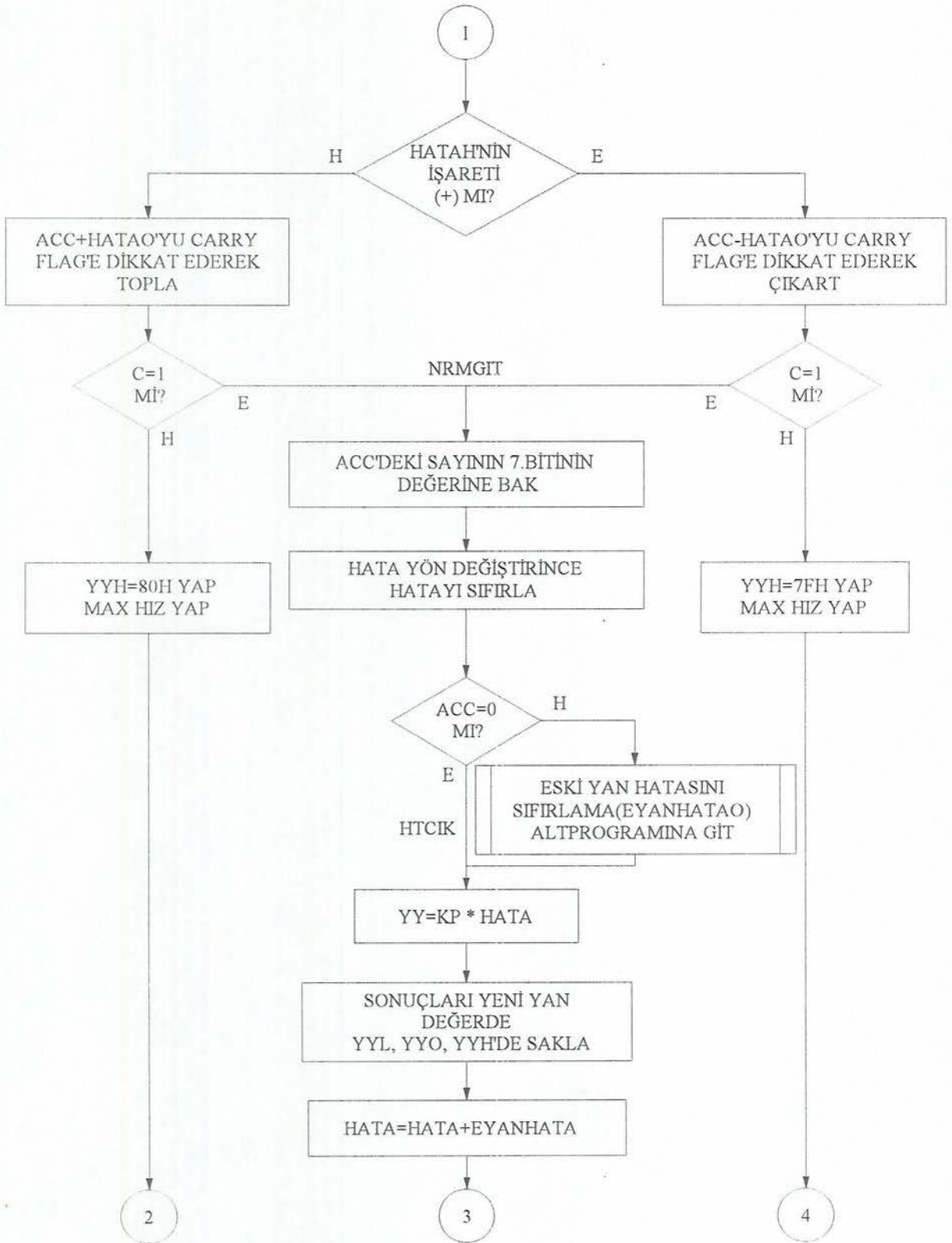


Şekil 4.4: Motor sürücü kartı devre şeması

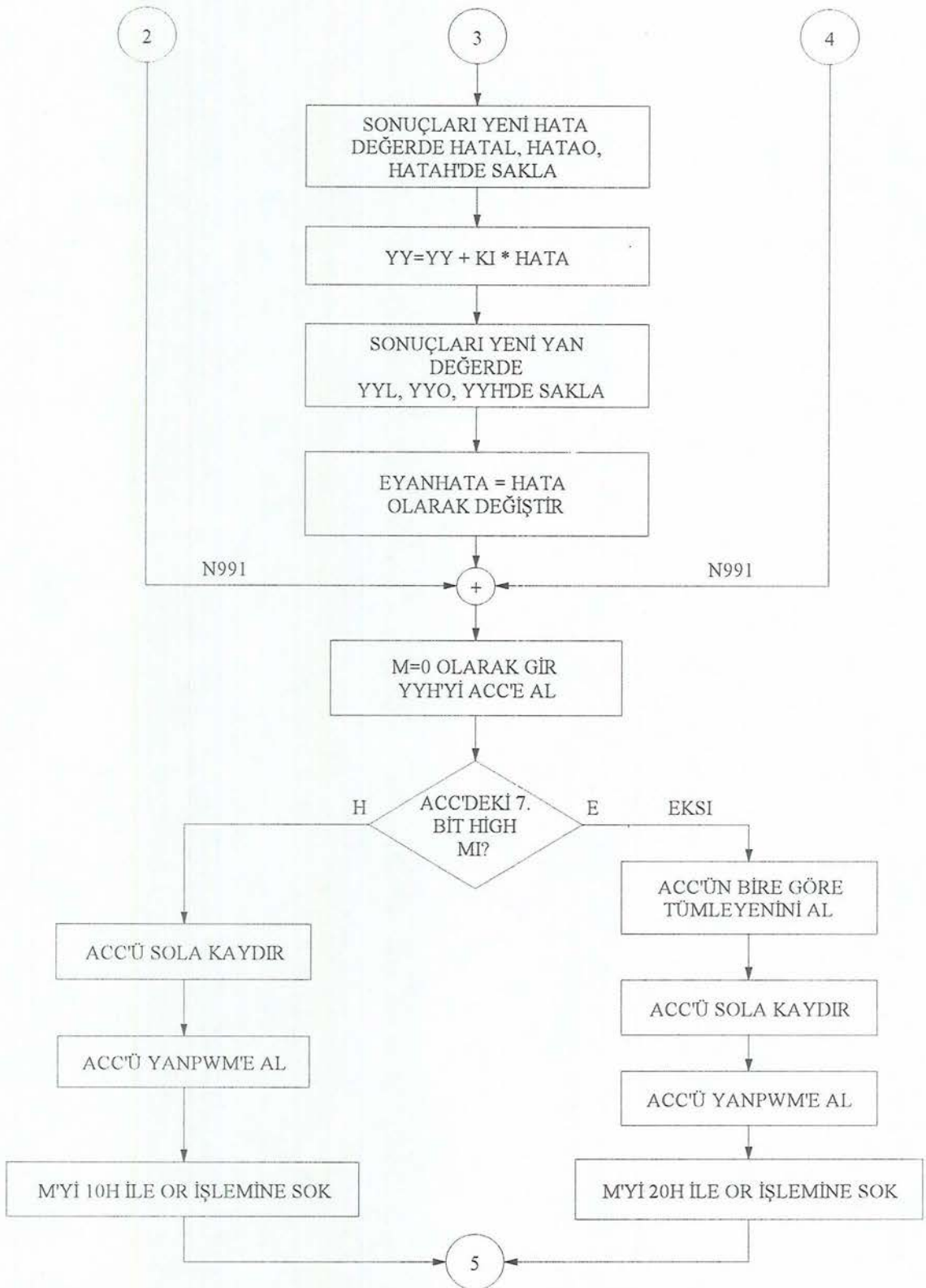
### 4.3. Kontrol Programı



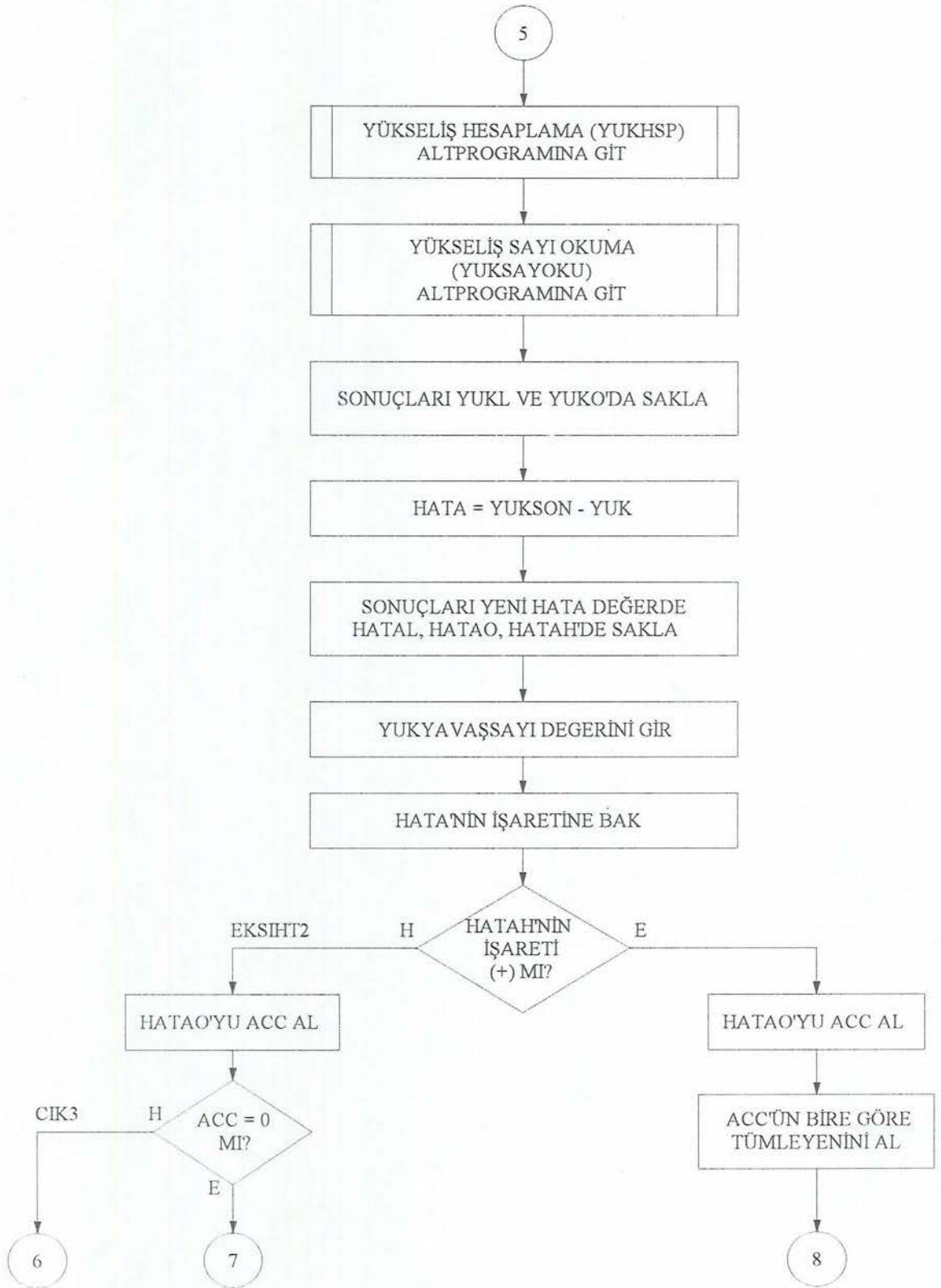
Şekil 4.5 : Kontrol programı akış diyagramı



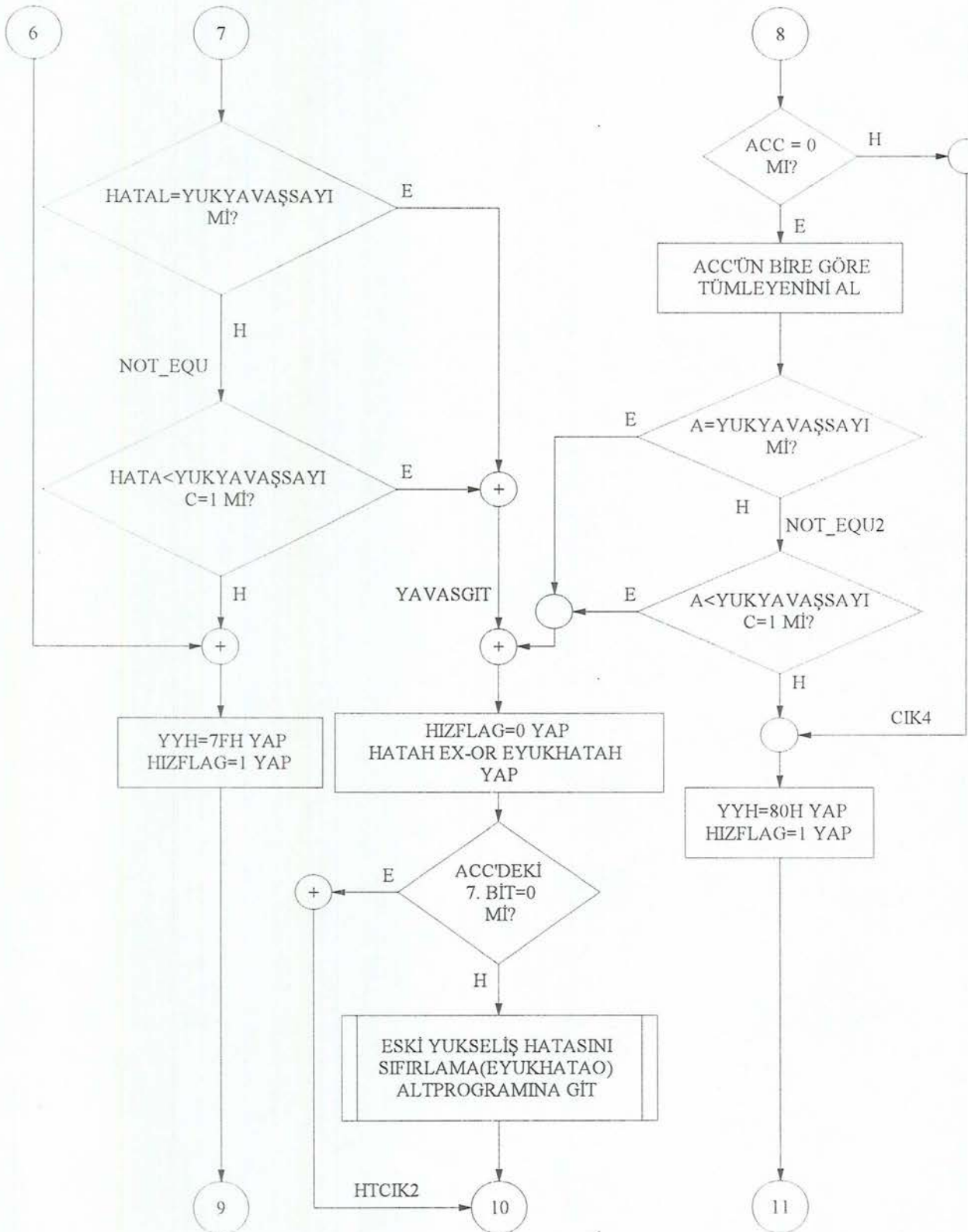
Őekil 4.5 : Kontrol programı akıő diyagramının devamı



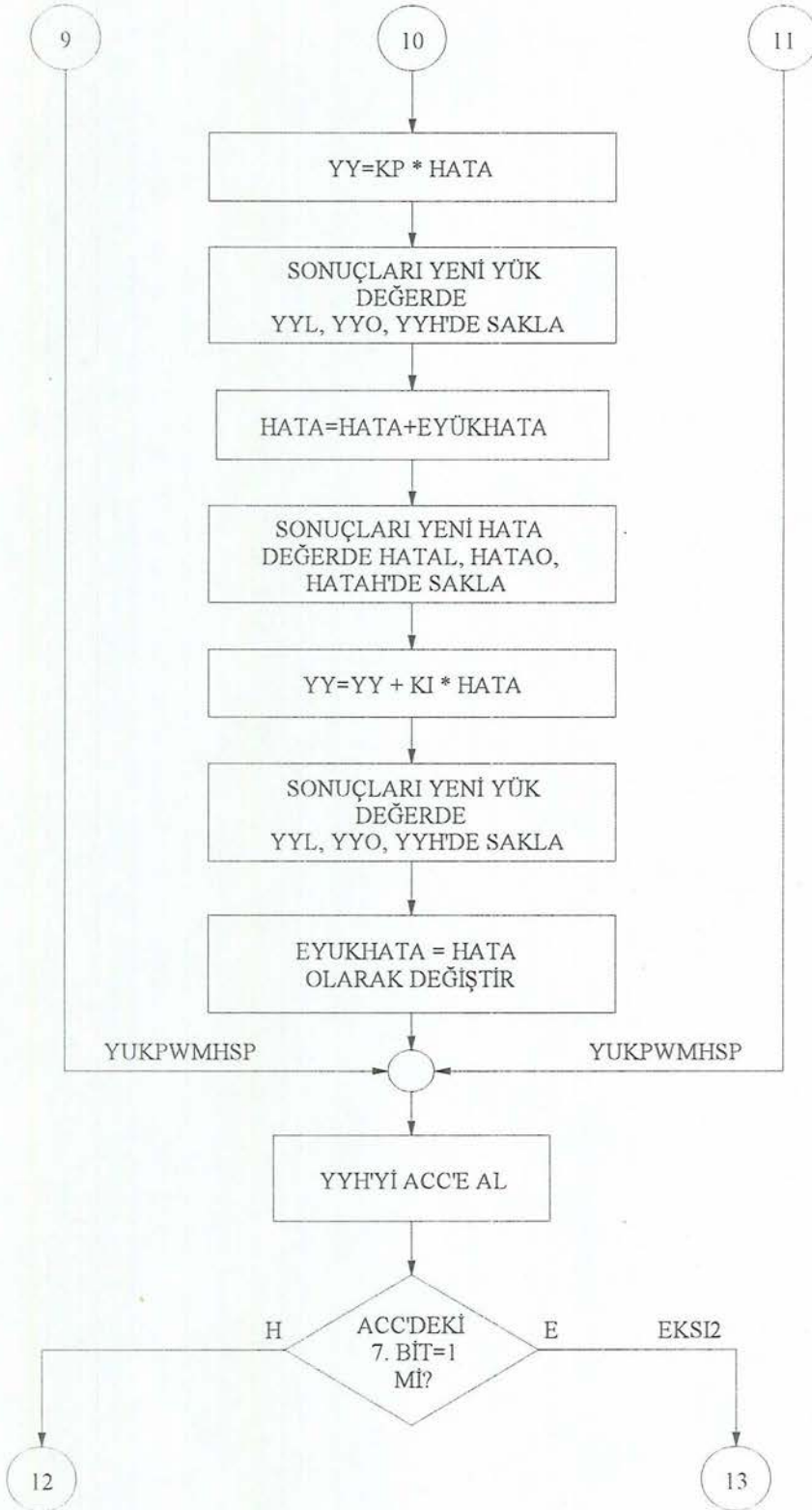
Şekil 4.5 : Kontrol programı akış diyagramının devamı



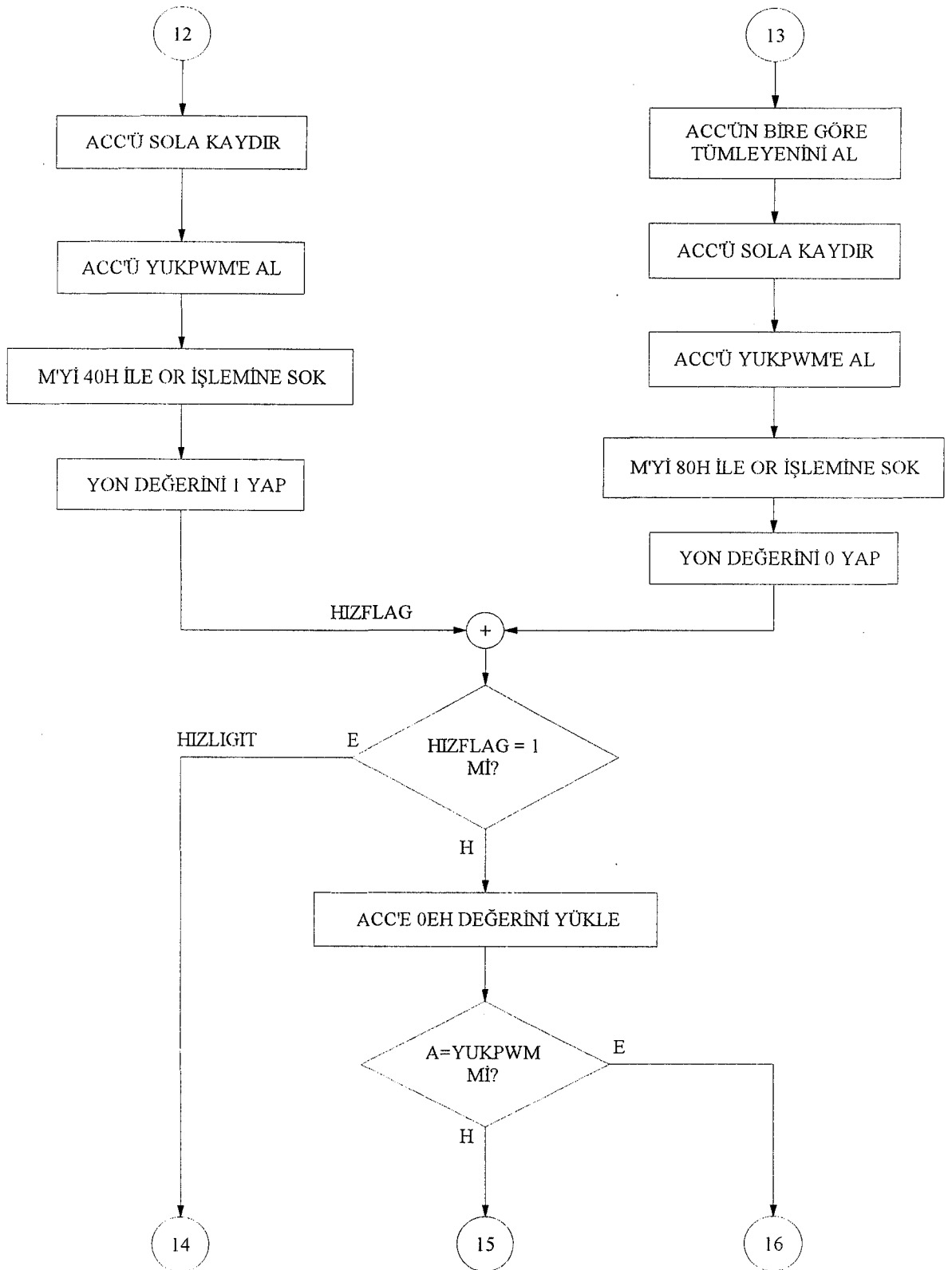
Şekil 4.5 : Kontrol programı akış diyagramının devamı



Şekil 4.5 : Kontrol programı akış diyagramının devamı

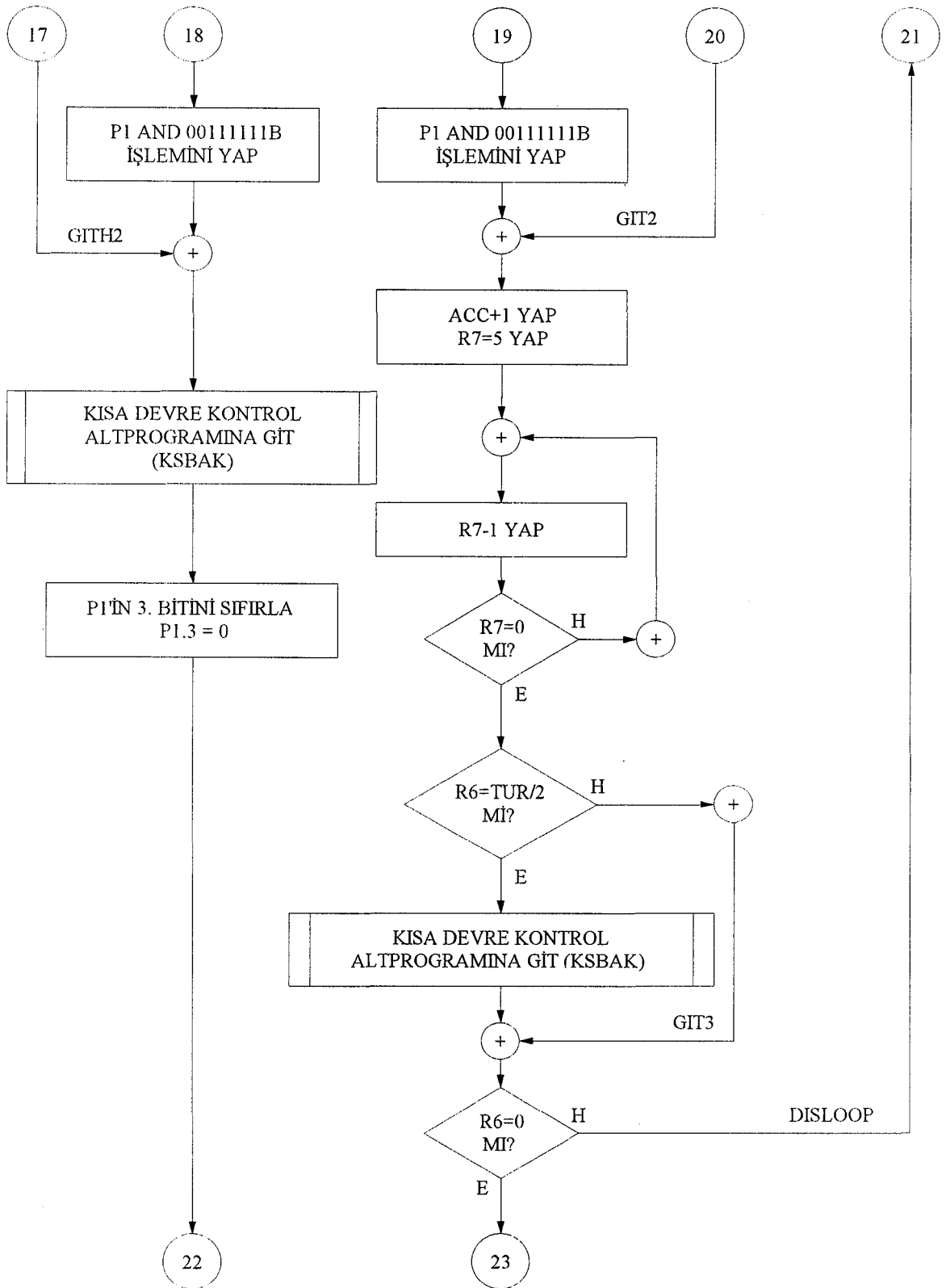


Şekil 4.5 : Kontrol programı akış diyagramının devamı

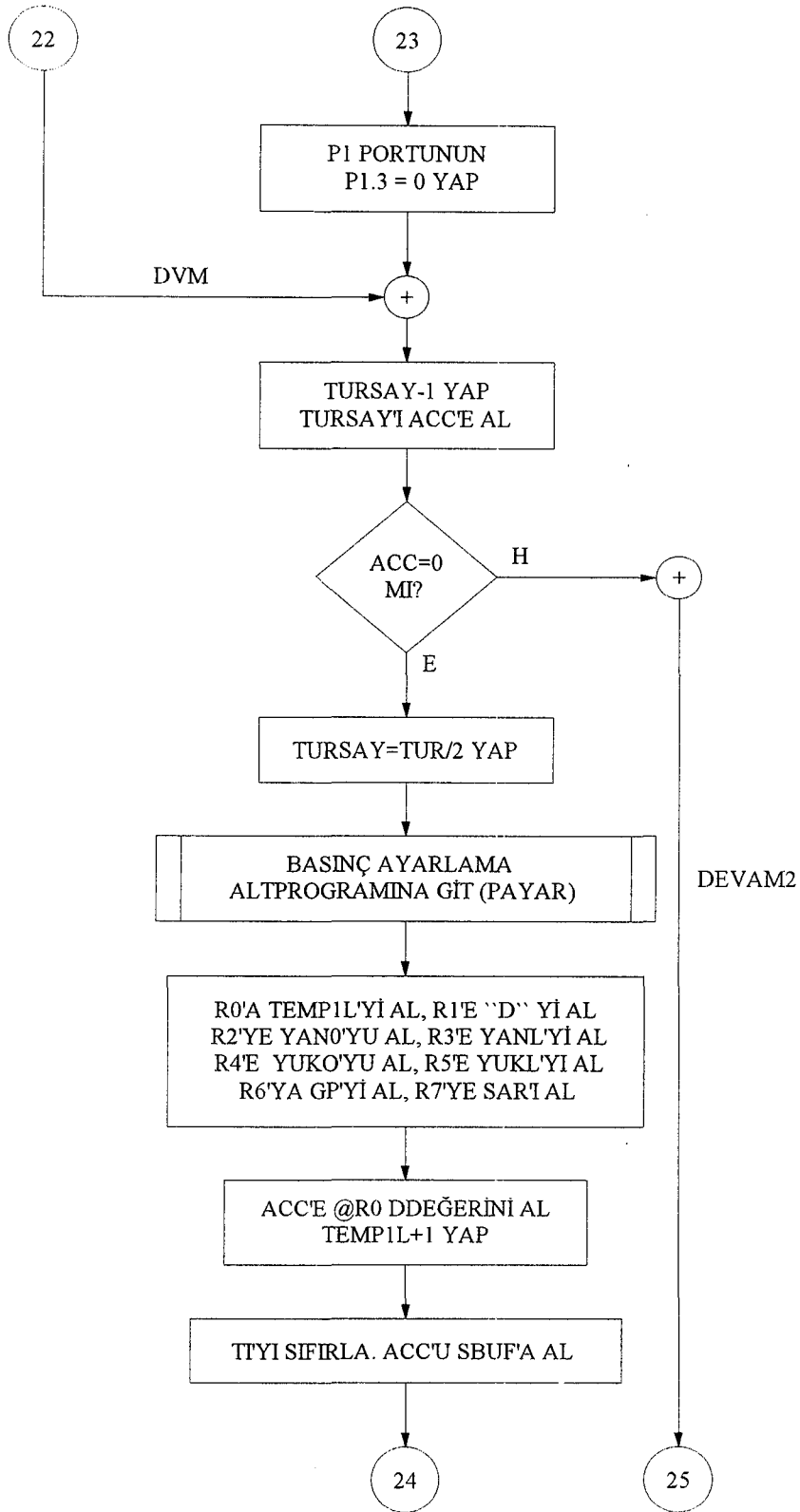


Şekil 4.5 : Kontrol programı akış diyagramının devamı

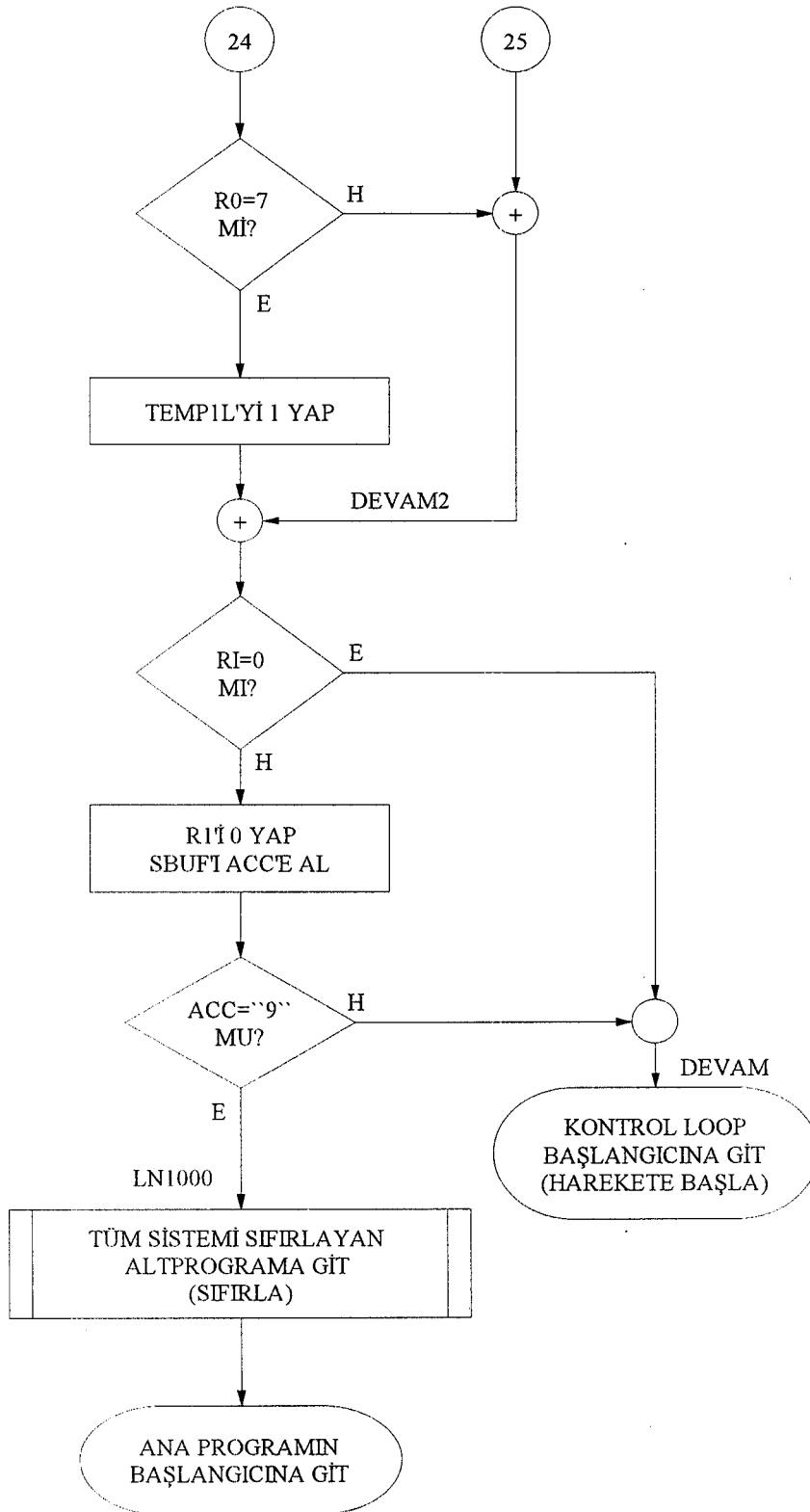




Şekil 4.5 : Kontrol programı akış diyagramının devamı



Şekil 4.5 : Kontrol programı akış diyagramının devamı



Şekil 4.5 : Kontrol programı akış diyagramının devamı

Etiket (label) HAREKETBASLA ile kontrol programına atlamak mümkündür. (Bakınız Ek 2) Bu etiketten sonra YANHSP etiketine giriş yapılır. YANHSP bölümünde ilk olarak YANSAYOKU isimli bir altprogram işletilir. YANSAYOKU altprogramında (Açıklama 1); P1 portu 1111 0000B sayısı ile lojik AND işlemine sokulur. Bunun nedeni sayım kartı üzerinde bulunan U3 (6273) entegresi, 8 kanal analog çoklayıcıda bulunan G0, G1, G2 ve EN kontrol girişlerinin seçilmesi gerekmektedir. (Bakınız Şekil 4.7) Lojik AND işlemi ile G0, G1 ve G2 kontrol girişleri lojik seviye olarak sıfır değerini alırlar. Buna bağlı olarak D0 çıkışı seçilmiş olur. D0 çıkışı U1 ve U2 (74HCT573) entegrelerinin OC bacağına bağlıdır. Bu bacak lojik 1 seviyesine çekildiği için U1 ve U2 entegrelerinin bilgi okuma işlemleri kesilir. Bu entegrelerde bilgi okuma işlemi kesildiği sıradaki değerler saklı kalacaktır. Sırasıyla Port B'de okunan değer, YANO (Yan değer orta baytı) adresine alınır. Daha sonra Port A'da okunan değer, YANL (yan değer düşük baytı) adresine alınır. Eğer sistemde bir offset değer oluşmuş ise eklenmesi gerekir. Yan offset değeri R0 ve R2 registerlerinde mevcuttur. YANSAYOKU altprogramı içinde IADD isimli başka bir altprogram çağrılır. IADD adlı bu altprogramda (Açıklama 2); R0 ve R2 registerleri içindeki sayılar R1 ve R3 registeri içindeki sayı ile toplanır. Toplama sonucu R1 ve R3 registerine yazılır. Programda kalınan yere geri dönülür.

Elde edilen YAN ( $YAN = YANL + 256 * YANO$ ) değerine, DYAN (yan olarak diferansiyel gitme miktarı) değeri eklenirse, en son YAN olarak gidilen noktanın koordinat değeri elde edilmiş olur. Bu değerde kısaca YANSON olarak adlandırılır.  $YANSON = YANSONL + 256 * YANSONO + 256 * 256 * YANSONH$  şeklinde ifade edilmektedir. Bu işlem yapılırken, IADD3 isimli bir altprogram kullanılır. IADD3 isimli altprogramda (Açıklama 3); R0, R2 ve R4 registerlerinin içindeki sayı R1, R3 ve R5 registerleri içindeki sayı ile toplanır. Toplama sonucu R1, R3 ve R5 registerleri içinde saklanır.

Daha sonra YUKSAYOKU isimli altprogram işletilir. (Açıklama 4) Bu altprogramda P1 portu 0000 0001B sayısı ile lojik OR işlemine sokulur. Bu işlem sonucunda P1 portunun P1.0. biti etkilenir. P1 portu 1111 0001B sayısı ile lojik AND işlemine sokulur. Bu işlemde G0 kontrol girişi daha önceki lojik OR işlemi ile lojik 1 seviyesine getirilmiştir. Bu kontrol sinyali lojik AND işlemi ile değeri değişmemiş olur. Fakat G1, G2 ve EN kontrol değerleri lojik 0 seviyesine getirilmiş olur. Buna benzer yapı Açıklama 1'de yapılmıştır. U3 entegresindeki kontrol girişleri G0: lojik 1, G1 ve G2: lojik 0 durumundadır. Bu durumda D1 çıkış anahtarı aktif olur. Bu D1 çıkışı U1 ve U2 entegrelerinin bilgi okumasını engelleyecektir. Bu entegrelerin çıkışında sabit bir değer oluşur. Port B'de okunan değer YUKO (yükseliş değeri orta baytı) adresine alınır. Port A'da alınan değer YUKL (yükseliş değeri düşük baytı) adresine alınır. Eğer sistemde bir offset değer oluşmuş ise bunun sisteme etki ettirilmesi gerekir. Bu sebepten dolayı R0 ve R2 registeri içinde saklı olan yükseliş offset değeri, YUKO ve YUKL değerine eklenir. Ana programa geri dönülür.

Elde edilen YUK ( $YUK = YUKL + 256 * YUKO$ ) değerine, DYUK (yükseliş olarak diferansiyel gitme miktarı) değeri eklenirse, en son YUK (yükseliş değeri) olarak gidilen noktanın koordinat değeri elde edilmiş olur. Bu değerde kısaca YUKSON olarak adlandırılmıştır.  $YUKSON = YUKSONL + 256 * YUKSONO + 256 * 256 * YUKSONH$  şeklinde ifade edilir.

TEMP10, TEMP20 değişken değerleri, kısa devre kontrolü için sıfırlanır. TEMP1L değişken değeri, değer yollama sayacı olarak kullanılacaktır. Bu değişken değeri 1 yapılacaktır.

Sisteme ait kontrol döngüsüne başlayabiliriz. İlk olarak yan motor hesabı yapılır. Bunun için YANHSP etiketine atlanır. YANSAYOKU adlı altprogram işletilir. Bu altprogram Açıklama 1'de ayrıntılı olarak açıklanmıştır. Bu altprogramda, en son okunan yan değerler elde edilir. Ana programa dönülür. Ana programda bu elde edilen yan değerler, olmasını istediğimiz yan değerlerle karşılaştırıp hata olup olmadığına bakılır. Hata mevcut ise hatanın yönü tespit edilir. İşlem olarak HATA =YANSON-YAN şeklinde ifade edilir. (HATA=HATAL+256\*HATAO+ 256\*256\* HATAH)

ISUB2C adlı altprogram, üç baytlık çıkarma işlemi için tanımlanmış özel bir altprogramdır.

Yan yaklaşma sayı değeri 77,58 milyem olarak belirlenir. HATAH'nin değeri Acc'e alınır. Acc'e alınan sayının 7. biti lojik 1 ise etiket EKSIHT'ye atlanır (hatanın işareti negatiftir), 7. bit lojik 0 ise işleme devam edilir. (hatanın işareti pozitifdir) Eğer hata pozitif ise HATAO'yu Acc'e alınır, carry flag temizlenir. Acc'den ödünç eldeye dikkat edilerek yan yaklaşma sayısı çıkartılır. Çıkartma işlemi sonucunda carry flag 1 ise NRMGIT (normal gitme) etiketine atlanır. Çünkü hata yan yaklaşma sayısından küçüktür. Carry flag 0 ise yeni yan değer yüksek baytı (YYH) 7FH değerini alır. Bu da (+) yönde maksimum hız demektir. Daha sonra N991 etiketine atlanır. Eğer hatanın işareti negatif ise HATAO'yu Acc'e alınır, Acc'ün bire göre tümleyeni bulunur. Carry flag sıfırlanıp, Acc'den ödünç eldeye dikkat ederek yan yaklaşma sayısı çıkartılır. Çıkartma işlemi sonucunda carry flag 1 ise NRMGIT (normal gitme) etiketine atlanır. Eğer carry flag 0 ise yeni yan değer yüksek baytı (YYH) 80H değerini alır. Bu da (-) yönde maksimum hız demektir. Daha sonra N991 etiketine atlanır.

NRMGIT etiketine gelindiğinde, yapılan hata istenen sınır değeri içindedir. Hata değerinin üst baytı (HATAH) Acc'e alınır. Eski yan hata değerinin yüksek baytı (EYANHATAH) Acc ile lojik exculsive-OR işlemine sokulur. Acc'deki sayının 7. bitine bakılır. 7. bit lojik 0 ise hatadan çıkma (HTCIK) etiketine atlanır. 7. bit lojik 1 ise eski yan hatayı sıfırlama (EYANHATAO) altprogramına dallanılır. Altprogramda eski yan hata düşük baytı (EYANHATAL), eski yan hata orta baytı (EYANHATAO), eski yan hata yüksek baytı (EYANHATAH) ve kısa devre kontrol değişkeni TEMP10 değerleri sıfırlanır. Ana programa geri dönülür. Hatadan çıkma (HTCIK) etiketine gelinir.

Hatadan çıkma etiketinde hata değişkenine PI kontrol uygulanır. İlk önce hata değişkeni Kp sabiti ile çarpılır. Bunun için R0 registerine Kp değeri alınır. R2 ve R4 registeri sıfırdır. R1, R3 ve R5 register içeriklerinde daha önceden HATA değerleri bulunmaktadır. IMUL3 adlı altprogram yardımıyla Kp\*HATA işlemi yapılmış olacaktır. Bu altprogram (Açıklama 5); R5, R3 ve R1 register değerleri ile R0 register değerini çarpar ve sonucunu R5, R3 ve R1 registerine yazar. Eğer taşma varsa ekler. Ana programa geri döner.

R1 registerinin içeriği yeni yan değer yüksek bayt (YYH), R3 registerinin içeriğini yeni yan değer orta bayt (YYO), R5 registerinin içeriği yeni yan değer düşük bayt (YYL) alınır. Yeni oluşan hata şu şekilde bulunur: HATA= HATA+EYANHATA bulunmuş olan hata değerini KI sabiti ile çarpılıp, daha önce yeni yan değer olarak bulunan değerle toplanır. Sonuç PI uygulanmış yeni yan değerde (YY) saklanır. YY=YYL+

256\*YYO+256\*256\*YYH olur. Eski yan hata değerinin içeriğine (EYANHATA), hata değerinin içeriği (HATA) konur ve saklanır.

Daha sonra N991 etiketine atlanır. (YANPWM hesabı) İlk olarak M (motor) değeri sıfırlanır. Yeni yan değer yüksek baytı (YYH) Acc'e alınır. Acc'deki sayının 7. bitinin değeri 1 ise sayı negatiftir. Dolayısıyla EKSI adlı etikete atlanır. Eğer 7. bitin değeri 0 ise kaldığı yerden devam edilir. Acc'ü sola kaydırılır. (A7,A0 olur.) Yeni oluşan Acc değerini YANPWM değerine alınır. Motor değeri 10H ile lojik OR işlemine sokulur. Dolayısıyla DC motor sürücü kartı üzerindeki Port 1.4 nolu bit haricindeki diğer bitler lojik 0 seviyesindedir. (Bakınız Şekil 4.8) Port 1.4 nolu bitin, lojik 1 olması motorun sola doğru hareket etmesini sağlar. Daha sonra, CIK995 etiketine atlanacaktır. EKSI etiketine gelindiğinde, Acc'ün bire göre tümleyeni alınır. Acc sola kaydırılır. (A7,A0 olur.) Yeni oluşan Acc değeri YANPWM değerine alınır. Motor değeri 20H ile lojik OR işlemine sokulur. Dolayısıyla DC motor sürücü kartı üzerindeki Port 1.5 nolu bit haricindeki diğer bitler lojik 0 seviyesindedir. Port 1.4 nolu bitin, lojik 1 olması motorun sağa doğru hareket etmesini sağlar.

CIK995 etiketine gelindiğinde yükseliş motor hesabı (YUKHSP) adlı bölüme atlanır. Bu bölüm YUKHSP adlı etiketten başlayarak işletilir. İlk olarak yükseliş motorunun enkoder sayı değerinin okunduğu altprogram (YUKSAYOKU) işletilir. Bu altprogram Açıklama 4'de ayrıntılı olarak anlatılmıştır. Bu altprogramda yeni yükseliş motoru enkoder değeri düşük baytı (YUKO) ve yükseliş motoru enkoder değeri orta baytı elde edilir. Ana programa geri dönülür.

Konsoldan, sistemin yükseliş olarak hangi konuma getirileceği girilmiş durumdadır. Girilmiş bu yükseliş değeri milyem büyüklüğündedir. Bu değer üç bayta bölünmüştür. Bu değer yazıldığı adresler şunlardır: YUKSONL (yükseliş motoru enkoder son değerinin düşük baytı), YUKSONO (yükseliş motoru enkoder son değerinin orta baytı), YUKSONH (yükseliş motoru enkoder son değerinin yüksek baytı). Bu değerlerle daha önce sistemin başında elde edilen yükseliş değerleri karşılaştırılır. Bu karşılaştırmaya göre yükseliş motorunun hareketine karar verilir. Bunu da işlem olarak HATA=YUKSON-YUK olarak ifade edilir. Burada çıkarma işlemi için tanımlanan ISUB2C adlı altprogramı kullanılır. Sonuçta oluşan hata değeri şu adreslerde saklanacaktır: HATAL (oluşan hatanın düşük baytı), HATAO (oluşan hatanın orta baytı), HATAH (oluşan hatanın yüksek baytı).

Yükseliş değerinin, servo motorun yavaşlama yaptığı, belirli bir değeri mevcuttur. Burada bu değer belirlenir. Yükseliş yavaşlama sayı değerine (YUKYAVASSAYI) 25D değeri verilir. Burada hata değerinin yavaşlama sayı değerinden büyük veya küçük olmasına göre motora hız verilir. İlk olarak HATAH'yi Acc'e alınır. Burada hatanın yönüne bakılır. Acc'deki 7. bitin işareti (-) ise EKSIHT2 adlı etikete atlanır. Burada motorun hareketi (-) yönde olur. Eğer Acc'deki 7. bitin işareti (+) ise işleme devam edilir. HATAO'nun değeri Acc'den alınır. Burada Acc'ün değeri sıfırdan farklı olup olmadığına bakılır. Sıfırdan farklı ise hata değeri 256'dan büyüktür. Bu yüzden ÇIK3 adlı etikete atlanır. Eğer hata değeri 256'dan küçük ise; R1 registerinde HATAL'nin (hatanın düşük baytı) değeri bulunur. Bu değer yükseliş yavaşlama sayı değeri ile karşılaştırılır. Eğer R1=YUKYAVASSAYI değerine eşit ise dallanma olmadan programa devam edilir. Eğer R1'deki sayı YUKYAVASSAYI değerinden farklı ise NOT\_EQU adlı etikete dallanma sağlanır. Dallanma yok ise; direkt olarak YAVASGIT (motora yavaş yavaş hareket ver-

me) adlı etikete atlanır. Eğer NOT\_EQU etiketine dallanırsa carry flag'in değerine bakılır. Carry flag set edilmiş ise  $R1 < YUKYAVASSAYI$  değerinden küçüktür. Bu yüzden YAVASGIT adlı etikete atlanır. Eğer carry flag set edilmemiş ise CIK3 adlı etikete atlanır, bu dallanmada durum  $R1 > YUKYAVASSAYI$  şeklindedir. Bu şartta; (+) yönde maksimum hızda motorun hareket etmesi istenir. Yeni yükseliş değerinin üst baytına (YYH) 7FH değeri konur. HIZFLAG (hızın büyüklüğünü gösteren flag) set edilir. Daha sonra YUKPWMHSP (yükseliş PWM hesabının yapıldığı bölüm) adlı etikete atlanır.

Eğer motorun hareketinin (-) yönde olması istenir ise; EKSIHT2 adlı etikete atlanır. HATAO'nun değeri Acc'e alınır. Acc'ün bire göre tümleyeni alınır. Elde edilen Acc'ün değerine bakılır. Acc'ün değeri sıfırdan farklı ise hatanın değeri 256'dan büyüktür. CIK4 adlı etikete dallanma yapılır. Acc sıfıra eşit ise HATAL değeri Acc alınır. Acc'ün bire göre tümleyeni alınır. Elde edilen  $Acc = YUKYAVASSAYI$ 'ya eşit ise dallanma yapılmaz. YAVASGIT adlı altprograma dallanır. Eğer eşit değil ise NOT\_EQU2 adlı etikete atlanır. Bu etikette carry flag'in değerine bakılır. Carry flag set ise  $Acc < YUKYAVASSAYI$  demektir, YAVASGIT adlı altprograma dallanılır. Eğer carry flag set edilmemiş ise CIK4 adlı etikete atlanır. Yeni yükseliş değerinin yüksek baytına (YYH) 80H (1000 0000B) değeri konur. Hızflag değeri set edilir. Daha sonrada YUKPWMHSP adlı altprograma atlama yapılır. YAVASGIT adlı altprograma atlanırsa; hızflag değeri temizlenir. HATAH değeri Acc alınır. Acc ile EYUKHATAH değerini lojik Exclusive-OR işlemine sokulur. Hatanın yön değiştirip değiştirmediği kontrol edilir. Acc'deki 7. bitin değerinin sıfır olup olmadığına bakılır eğer sıfır ise HTCİK2 (hatadan çıkma) adlı etikete atlanır. 7. bit 1 ise EYUKHATAO adlı altprograma dallanır ve hata değerleri sıfırlanarak HTCİK2 adlı etikete geri dönülür. Bu bölümde hata değişkenlerine PI kontrol uygulanır. İlk önce HATA değeri  $K_p$  sabiti ile çarpılır. Yan motorun hareketi sırasında da aynı işlemler yapılmıştır. (Açıklama 5)

Sırasıyla  $YY = K_p * HATA$ ,  $HATA = HATA + EYUKHATA$  ve  $YY = Y + K_i * HATA$  şeklinde işlemler dizini uygulanır. Daha sonra EYUKHATA = HATA olarak değiştirilir. YUKPWMHSP adlı altprogram işletilir. Yeni yükseliş üst bayt değeri Acc'e alınır. Acc'deki 7. bit lojik1 ise EKSI2 adlı etikete dallanma sağlanır. Acc'deki 7. bit lojik 0 ise yükseliş yön hareketi (+) yöndedir. Acc'ü bir bit sola kaydırılır. Acc'ün değerini YUKPWM'e alınır. DC motor sürücü kartında Port 1, motorların hareket yönünü tayin etmektedir. Port 1'in 4, 5, 6 ve 7 nolu bağlantı hatlarından, 4 ve 5 nolu bağlantı hatları yan motor kartına, 6 ve 7 nolu bağlantı hatları yükseliş motor kartında yön tayin edecek şekilde tasarlanmıştır. Motor değerine 40H sayısı yazılırsa; yani Port 1, 0100 0000B şekline getirilmiş demektir. Bu durumda Port 1'in 6 nolu bağlantı hattı lojik 1 seviyesinde olur. Bu da yükseliş motorunun sola doğru dömesini sağlar. Bu dönüş yönünü belirtmek için yön adres değeri 1 yapılır ve ana programda kalınan yere geri dönülür.

Eğer program EKSI2 adlı etikete dallanmış olsaydı, ilk olarak Acc'ün bire göre tümleyeni alınır, (çünkü Acc'deki sayı negatiftir.) daha sonra Acc'ü sola kaydırılır ve artık Acc'deki sayı işaret bitinden arındırılmış oldu. Acc'deki sayıyı YUKPWM'e (yükseliş PWM değeri) alınır. Motor değerine 80H sayısı yazılır ise yani Port 1, 1000 0000B şekline getirilmiş olur. Bu durumda Port 1'in 7 nolu bağlantı hattı lojik 1 seviyesinde olur. Bu da yükseliş motorunun sağa doğru dömesini sağlar. Bu dönüş yönünü belirtmek için yön adres değeri 0 yapılır ve ana programda kalınan yere geri dönülür.

Hızflag bitinin lojik 1 olup olmadığına bakılır. Eğer bit lojik 1 ise HIZLIGIT adlı etikete atlanır. Eğer bit lojik 0 ise işleme aynı yerden devam edilir. Acc'e 0E0H (1110 0000B) değeri alınır. A=YUKPWM'e eşit olup olmadığı kontrol edilir. Eşit değil ise SOR adlı etikete gidilir. A<YUKPWM şeklinde ise carry flag set edilir. A>YUKPWM şeklinde ise carry flag temizlenir.

Programda SOR adlı etikete gelinir ise; carry flag'in durumuna bakılır. Carry flag set edilmiş ise program HIZLIGIT adlı etikete dallanır. Aksi halde servo motorlarda, istenen değere yavaş yaklaşma söz konusu olacaktır. P1 portu M (motor bilgisi) yapılı ve Port 1'in 3. biti set edilir, böylece bus boşaltılmış olur. R6 registerine 255D, R0 registerine 00D değeri konur, Acc temizlenir ve DISLOOP adlı bir döngü oluşturulur.

Döngüdeki mevcut durumun; Acc=YANPWM'e eşit olup olmadığına bakılır. Eşit değil ise GIT1 adlı etikete dallanır. Eşit ise Port 1, 1100 1111B sayısı ile lojik AND işlemine sokulur. Bu durumda yan motoru süren pulslar sona erdirilir. GIT1 adlı etikete dallanma yapılırsa; Acc=YUKPWME eşit olup olmadığına bakılır. Eşit değil ise GIT2 adlı etikete dallanma sağlanır. Eşit ise Port 1, 0011 11111B sayısı ile lojik AND işlemine sokulur. Bu durumda yükseliş motorunu süren pulslar sona erdirilmiş olur.

GIT2 adlı etikete gelindiğinde Acc'ün içeriği bir arttırılır. R7 registerinin içeriğine 05D sayısı konur. R7'nin içeriği bir azaltılır ve R7'nin içeriğinin sıfır olup olmadığına bakılır. R7'nin içeriği sıfır olana kadar aynı yerde kalınır. R6 registerinin içeriğinin, tur sayısının yarısına eşit olup olmadığına bakılır. (TUR/2) Eşit değil ise GIT3 adlı etikete dallanılır. Eşit ise; kısa devre kontrolü için, KSBK (kısa devreye bakma) altprogramına atlanır. Bu program yan ve yükseliş motorlarına ait kısa devre kontrolünün yapıldığı bir altprogramdır.

GIT3 adlı etikete gelindiğinde R6 registerinin içeriğinin sıfır olup olmadığına bakılır. Eğer sıfır değil ise DISLOOP etiketine tekrar dönülür ve aynı işlemler tekrarlanır. Port 1'in 3. biti temizlenir ve bus'ı boşaltma sona erdirilir ve DVM adlı etikete atlanır.

DVM adlı etikete gelindiğinde tur sayma değerinin içeriği bir azaltılır. (TUR SAY-1) TURSAY'ın içeriği Acc'e alınır. Acc'ün içeriğine bakılır. Acc'ün içeriği sıfır değil ise DEVAM2 adlı etikete atlanır. Acc'ün içeriği sıfır ise her tur pulsta bir buraya gelmesi için TUR/2'nin değerinin, TURSAY adlı değerinin içeriği olarak kayıt edilir. Basınç ayarlarının kontrol edildiği, PAYAR adlı altprograma dallanılır.

R1 registerine 'D' harfini, R2 registerine YANO (yan değer orta baytı), R3 registerine YANL (yan değer düşük baytı), R4 registerine YUKO (yükseliş değer orta baytı), R5 registerine YUKL (yükseliş değer düşük baytı), R6'ya gerçek basınç değeri, R7'ye arıza durum numarası (SAR), R0'da değer yollama sayaç değeri yüklenir.

Acc'e R0'ın içeriğini alınır. Değer yollama sayacının içeriği bir arttırılır. Timer işkesme registerini sıfırlanır. Acc'ün içeriğini seri tampona (SBUF) alınır. R0 registerinin, 7'ye eşit olup olmadığı kontrol edilir. (Değer yollama sayacı olarak kullanılıyor) Eşit değil ise DEVAM2 adlı etikete dallanılır, eşit ise değer yollama sayacı sıfırlanır ve DEVAM2 adlı etiket ile programa devam edilir. Register işkesmesi RI = 0 olup olmadığına bakılır. Eşit ise DEVAM adlı etikete dallanılır, eşit değil ise register işkesmesi sıfırlanır. Seri tampona (SBUF) Acc'ün içeriği alınır. Acc = 9 eşit olup olmadığına bakılır. Eğer eşit ise 1000 nolu program satırına dallanma sağlanır. Eğer eşit değil ise DEVAM adlı etikete dallanma yapılı. Bu dallanılan etikette, BASLA adlı etikete dallanma olur. (Kontrol loop başlangıcına) 1000 nolu program satırına dallanma sağlandığında tüm sistem

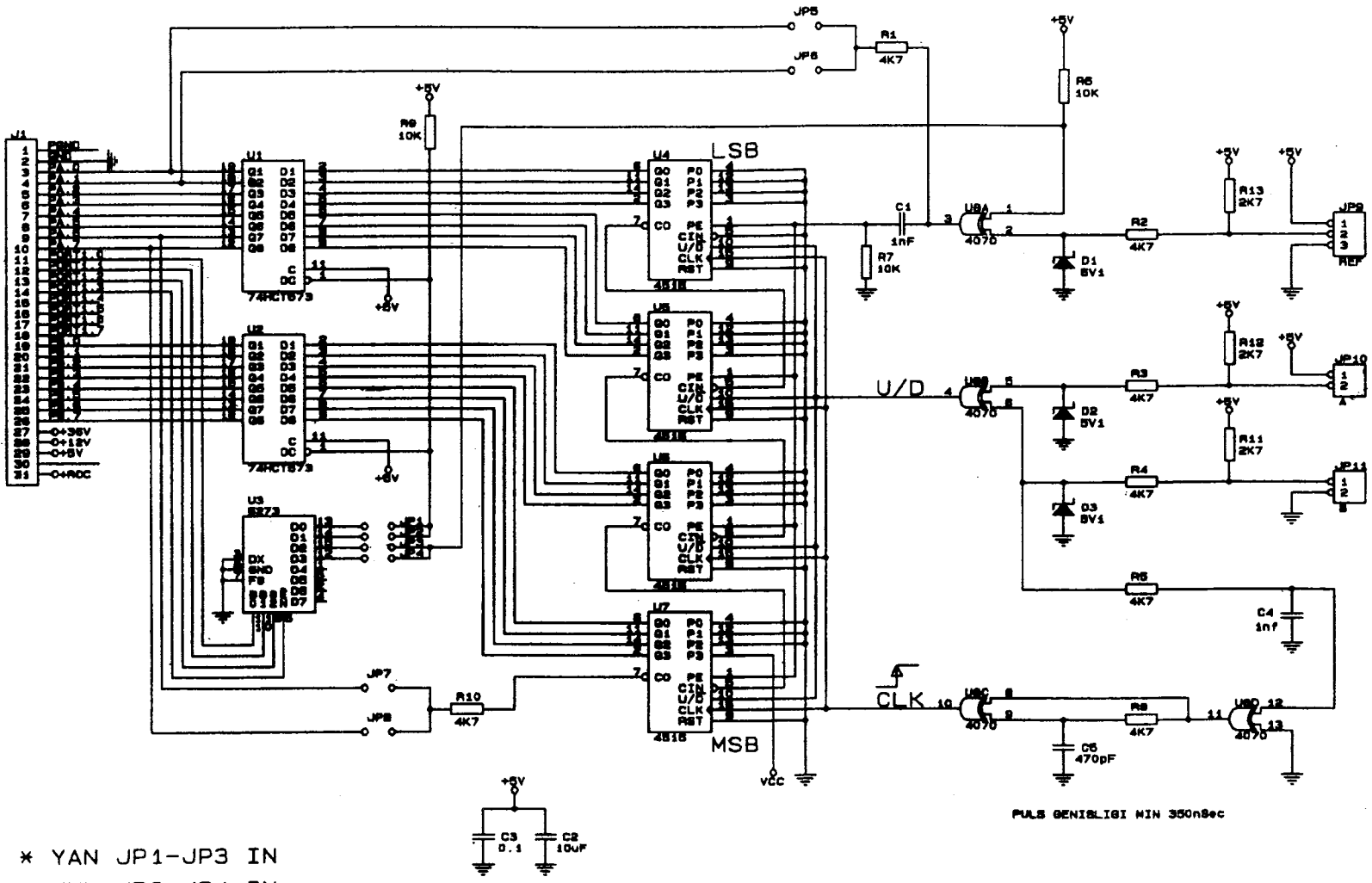
değerlerini sıfırlayan SIFIRLA adlı altprograma dallanma sağlanır. Bu durumda ana programın başına dallanma sağlanmış olur.

Eğer HIZLIGIT adlı etikete dallanma olursa; motor değerinin içeriği (M) Port 1'e alınır, Port 1'in 3. biti bus'u boşaltmak için set edilir. YANPWM değeri Acc'e alınır. Acc'ün içeriğine bakılır Acc=0 değil ise GITH1 adlı etikete dallanma gerçekleşir. Acc=0 ise Port 1, 1100 1111B değeri ile lojik AND işlemine sokulur. Bu durumda yan motoru hareket ettiren 4 ve 5 nolu bağlantı hatlarının lojik seviyesi sıfıra çekilir.

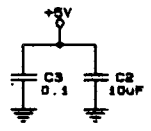
GITH1 adlı etikete dallanıldığında, YUKPWM Acc'e alınır. Acc'ün içeriğini kontrol edilir. Acc'ün içeriği sıfır ise Port 1'i, 0011 1111B değeri ile lojik AND işlemine sokulur. Bu durumda yükseliş motoruna hareket veren 6 ve 7 nolu bağlantı hatlarının lojik seviyesi sıfıra çekilir. Acc'ün değeri sıfırdan farklı ise GITH2 adlı etikete dallanma olur. İlk önce motorlarda herhangi bir kısa devre olup olmadığını kontrol eden, KSBK adlı altprograma dallanma sağlanır. Port 1'in 3. biti lojik 0 seviyeye çekilir. Programa, DVM adlı etiketten başlanarak devam edilir.



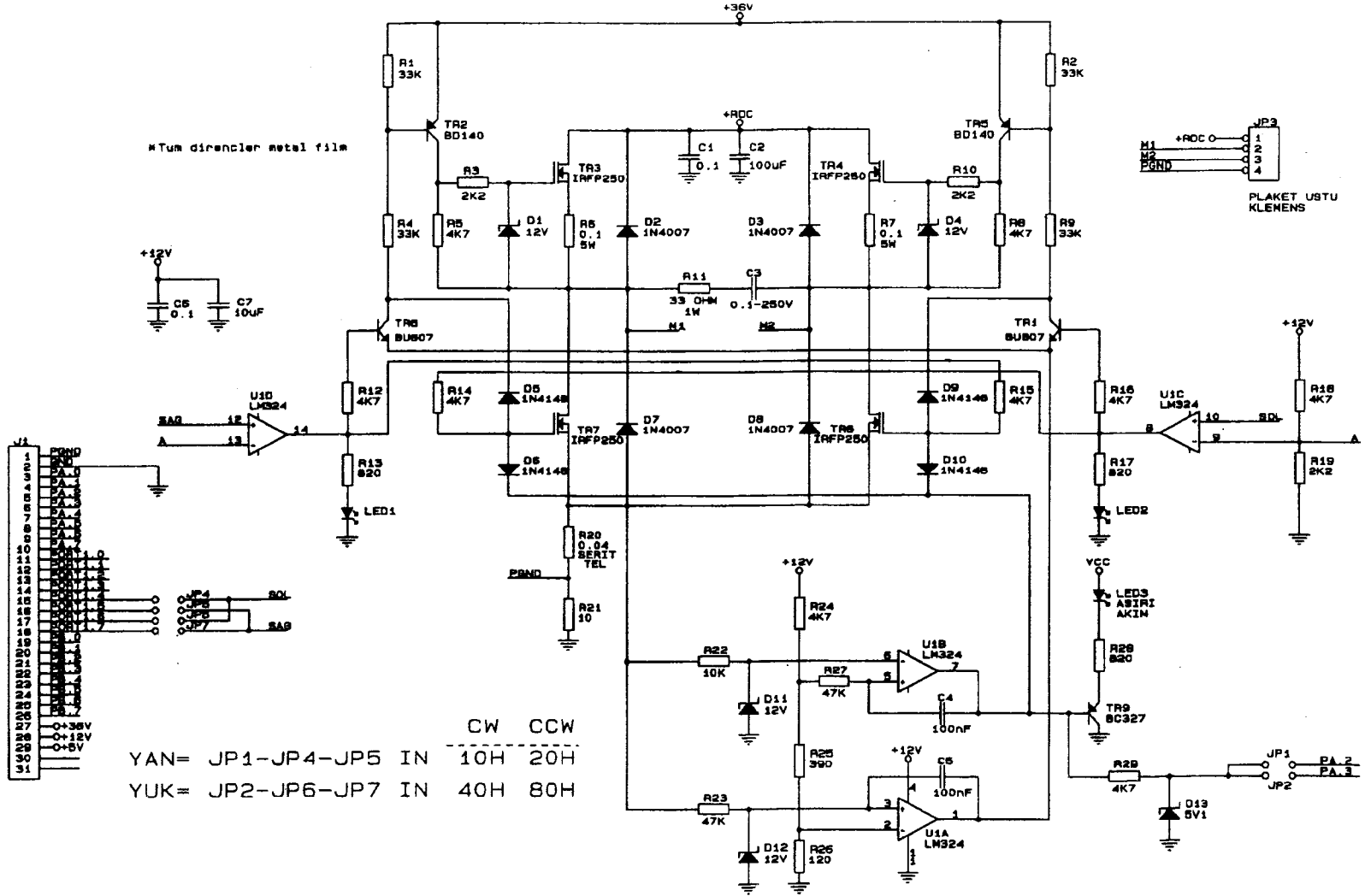
Şekil 4.7: Enkoder sayım kartı devre şeması



- \* YAN JP1-JP3 IN
- \* YUK JP2-JP4 IN
- \* JP5-JP6-JP7-JP8 TAKILMAYACAK.



Şekil 4.8: PWM motor kontrol sürücü kartı devre şeması



## 5. BİLGİSAYARLI ATIŞ KONTROL SİSTEMİNİN İŞLETİMİ

### 5.1. Sistemin Teknik Özellikleri

Besleme Gerilimi	: 24 VDC (Akü Beslemeli) (16V -30V)
Akım	: Eksen Başına Max. 20A
Güç	: Eksen Başına Max. 500W
Motor Tipi	: 24 VDC Servo - 1.1 Nm 0.5 ms
Kontrol	: Konsoldan Tuş Takımı veya telsiz BAIKS iletişimi
İklim Koşulları	: -40°C / +70°C, %95 Bağıl Nem

### 5.2. Sistemin Ana Elemanları

Bilgisayarlı Atış Kontrol sistemi (BAK-52) üç ana bölümden oluşmaktadır. Şekil 5.1'de sistemin blok şeması verilmiştir, bu bölümler şunlardır:

- i. Konsol Ünitesi(Keyboard ve Display)
- ii. Kontrol Ünitesi
- iii. Duyargalar (Eğim ölçer, basınç ölçer, enkoder)

#### 5.2.1. Konsol Ünitesi

Konsol BAK-52 sisteminin ana kumanda kutusu olup dokunmatik tip tuş takımı, sıvı kristal (LCD) ekran ve bilgisayar ünitesinden meydana gelmiştir ve sistemin tüm fonksiyonlarının kontrolü buradan yapılmaktadır.

BAK-52 sistemine verilecek tüm komut ve kontroller ile sistemin pozisyonunu gösteren ekran konsol üzerine monte edilmiştir. Konsol aynı zamanda Kontrol Ünitesindeki bilgisayar ile sürekli haberleşerek sistemin çalışırılığını ve yan birimlerde meydana gelebilecek herhangi bir arızayı ekranda yazılı mesaj ile operatöre bildirir. Aynı zamanda BAIKS' den gelen bilgiler de konsol tarafından değerlendirilir ve uygulamaya alınır.

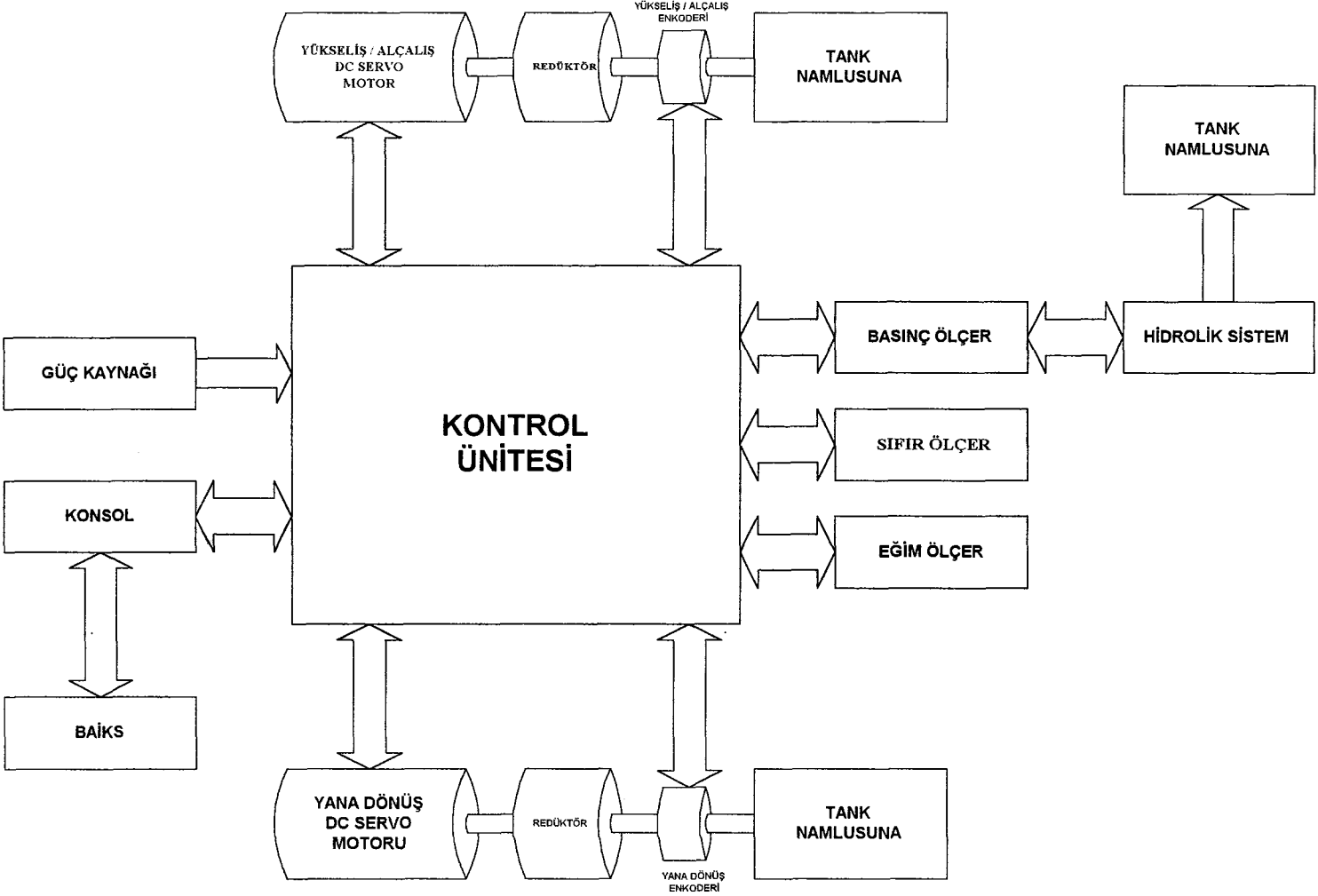
Ekranda denge basıncı "Bar", diğer tüm değerler "milyem" olarak gösterilmiştir.

#### 5.2.2. Kontrol Ünitesi

Kontrol ünitesi elektronik yapı olarak 2 adet motor sürücü, 2 adet eksen değer sayacı, 1 adet ölçme ve kontrol, 1 adet bilgisayardan oluşmuştur. Sistemin yan birimleri olan Servo motorlar, Enkoderler, X/Y Eğim Ölçer, Basınç Ölçer ve Denge basıncı valfleri bu üniteye soketler ile bağlanmıştır.

Buradaki bilgisayar Konsol bilgisayarından gelen bilgileri değerlendirip ilgili motor sürücülerini çalıştırır. Denge basıncı valflerini kontrol eder. Aynı zamanda eksenlerin hareket durumuna göre enkoderlerden eksen değer sayıcılarına gelen bilgileri, X-Y eğim değerlerini ve denge basınç değerini konsol bilgisayarına gönderir. Konsol besleme geriliminide buradan almaktadır.

Şekil 5.1. Sistemin blok şeması olarak gösterimi



### 5.3. Sistemin Çalıştırılması

OBÜS'ün veya tankın kontak anahtarının 1. kademeye alınması ile kontrol kutusuna ve diğer yan birimlere elektrik gitmiş durumdadır. Konsol ve sürücü çıkış ünitelerine elektrik vermek için kontrol kutusundaki sigortayı (F1) açınız ve konsol ekranını takip ediniz.

Sistem selftest (kendi kendine kontrol) özelliğine sahip olduğu için gerekli olan bölümleri test eder ve en son TAMAM mesajını yazarak ekran Şekil 5.2'deki düzeni alır.

BAK52	Yan	yuks
F1 ATIS	: 0000	: 0000
F2 ATMD	: 0000	: 0000
F3 NAMLU		
ACISI	: 3435	: 0465
MY. D. A.	:	
-----		
Diğer Menü İçin <CIK>		

Şekil 5.2. Sistemin ilk çalıştırılması sırasındaki konsol ekran görüntüsü

ATIŞ ve ATMD yan yükseliş değerlerinin 0000 olmasına rağmen NAMLU AÇISI yan ve yükseliş değerlerinin ne olduğu bu noktada önemli değildir. Çünkü henüz mekanik sıfırlama işlemi yapılmamıştır. Ayrıca henüz herhangi bir hedef koordinat yazılıp uygulamaya konmadığından MY. D. A. yani **Meyil Düzeltilmiş Aç**ı değeri hesaplanıp yazılmamıştır.

Eğer ekranda herhangi bir uyarı veya arıza mesajı var ise mesajın gereğini yapınız. Bunun için 5.6'daki Bakım veya 5.7'deki Onarım ve Arıza Tespit bölümüne geçiniz.

### 5.4. Sistemin İşletilmesi

Sistemin kullanımında yapılması gereken ilk uygulama mekanik sıfırlama ve aracın duruş pozisyonu itibarıyla eğim sıfırlama işleminin yapılmasıdır. Bunun için kule'yi çevirme kolu vasıtasıyla 3200 milyem olarak kabul edilecek yan açığa, namluyu da çevirme kolu vasıtasıyla yer düzlemine tam paralel olacak konuma getiriniz. (Yüks=0000). Konsol ekranında kursör "CIK" yazısı üzerinde durmaktadır. Tuş takımındaki "CIK" tuşuna basıp diğer menü'ye geçiniz. Diğer (ikinci menü) menü Şekil 5.3'deki gibidir.

<b>BAK52</b>	
<b>F1</b>	<b>EGİM SIFIRLAMA</b>
<b>F2</b>	<b>NAMLU ACISI</b>
	<b>SIFIRLAMA</b>
<b>F3</b>	<b>ELLE KUMANDA</b>
<b>F4</b>	<b>SİSTEM TEST</b>
<b>F5</b>	<b>ATIS BİLGİ</b>
<b>F6</b>	<b>TERCİHLER</b>
<b>F7</b>	<b>EL CARKI KNT.</b>
<b>D i g e r M e n u İ c i n &lt;CIK&gt;</b>	

Şekil 5.3. Sistemin işletilmesi sırasındaki konsol ekrandaki ikinci ana menü

<b>BAK52</b>	<b>yan</b>	<b>Yuks</b>
<b>F1 ATIS</b>	<b>: 0000</b>	<b>  0000</b>
<b>F2 ATMD</b>	<b>: 0000</b>	<b>  0000</b>
<b>F3 NAMLU</b>		<b> </b>
<b>ACISI</b>	<b>: 3200</b>	<b>  0000</b>
<b>MY. D. A.</b>	<b>:</b>	<b> </b>
-----		
<b>D i g e r M e n u İ c i n &lt;CIK&gt;</b>		

Şekil 5.4. Sistemin namlu açısı sıfırlama bölümü konsol ekran görüntüsü

F1 tuşuna basıp kursör'ün EGİM SIFIRLAMA satırına geldiğini görünüz ve YAP/EVET tuşuna basınız. Bu uygulamanın açıklaması takip eden bölümde yapılmıştır. Daha sonra F2 tuşuna basıp kursör'ün NAMLU ACISI SIFIRLAMA satırına geldiğini görünüz ve YAP/EVET tuşuna basınız. Sistem otomatik olarak Şekil 5.4'deki ekran düzenini almıştır.

Görüldüğü gibi NAMLU ACISI satırında Yan=3200, Yüks=0000 olmuştur. Bundan sonra verilecek tüm koordinat değerleri bu referans noktasına göre gerçekleştirilecektir. Ancak, eğer istenirse kule'nin bu referans noktası değiştirilebilir.

Artık sisteme hedeflenen koordinat değeri girilip kule ve namlu'nun kontrolü sağlanabilir. Bunun için F1 tuşuna basıp kursör'ün ATIS satırına geldiğini görünüz. Sağ yan ok tuşu (-->) ile yan değerinin bulunduğu haneler üzerine gelip istediğiniz yan koordinatını, yükseliş değerinin bulunduğu haneler üzerine gelip istediğiniz yükseliş koordinatını tuş takımını kullanarak yazıp YAP/EVET tuşuna basınız. Gereken denge basınç ayarı yapıldıktan sonra kule ve namlu hareket etmeye başlamıştır ve istenen koordinata geldiğinde otomatik olarak duracak, kursör "CIK" yazısı üzerine gelecektir.

Örnek :

Yeni koordinat değeri :  
                                     Yan    Yüks  
 F1 ATIS                  3300  0350

YAP/EVET tuşuna basıldıktan ve hareket tamamlandıktan sonra ekranın son hali Şekil 5.5'deki gibi olacaktır. Ayrıca bu uygulamada eğim sıfırlama işlemi yapıldıktan sonra eğim değişimi olmadığı kabul edilmiştir. Bundan dolayı MY. D. A. değeri verilen hedef koordinat değeri ile aynıdır.

<b>BAK52</b>		<b>yan</b>	<b>Yüks</b>
<b>F1 ATIS</b>	:	<b>3300</b>	<b>0350</b>
<b>F2 ATMD</b>	:	<b>0000</b>	<b>0000</b>
<b>F3 NAMLU</b>			
<b>ACISI</b>	:	<b>3300</b>	<b>0350</b>
<b>MY. D. A.</b>	:	<b>3300</b>	<b>0350</b>
-----			
<b>Diger Menu Icin &lt;CIK&gt;</b>			

Şekil 5.5. Sistemin hareketini tamamladıktan sonraki konsol ekran görüntüsü

Bu uygulamada yan değeri 2100 - 4800 arası, yükseliş değeri (-88) - 1155 arası verilebilir. Bunların dışındaki değerleri uygulamaya almaz ve SINIR DIŞI DEĞER yazılı mesajı ile operatöre bildirir.

Ayrıca ATMD olarak adlandırılan satırda yan ve yükseliş yazılacak (+) veya (-) değer, hedef koordinat değeri üzerine otomatik olarak eklenir veya çıkartılır. ATMD satırına erişim için F2 tuşu kullanılır.

NAMLU ACISI satırında görülen gerçek kule ve namlu koordinatları eğer istenirse buradan da sıfırlanabilir veya değiştirilebilir. Bunun için F3 tuşuna basınız. Kursör bu satıra gelecektir. Bu noktada olmasını istediğiniz yan ve yükseliş değerini

yazıp YAP/EVET tuşuna basınız. Bilgisayar bu işlemten sonra kule ve namlu'nun gerekli koordinat değerini yazılan yeni değerler olarak kabul edecektir.

Yukarıdaki uygulamada veya bundan sonra denenecek başka uygulamalarda platform eğiminde herhangi bir değişme olsaydı, bu değişim farkı ilk eğim sıfırlama işleminde referans olarak alınan eğim değerlerinden çıkartılarak aradaki fark miktarı kadar yan ve yükseliş değerleri yeniden hesaplanıp varılacak olan nokta NAMLU ACISI ve MY. D. A. satırına yazılacaktı. Namlu, bu hesaplanan noktaya varıp duracak olmasına rağmen namlunun deniz seviyesine göre mutlak açısı F1 ATIS satırına yazılmış olan koordinat değeri olacaktır.

Ayrıca bu menüde eğer silah sistemine mermi yüklenmiş ise F6 tuşuna basınız ve ekranda MERMİ DOLU yazılı mesajını görünüz. Bundan sonra sistem denge basıncı ayarını mermi ağırlığını da hesaba katarak yapacaktır. Eğer silah sistemine mermi yüklenmeden namlu hareket ettirilecekse tekrar F6 tuşuna basıp yazılı mesajın silinmesini sağlayınız.

Sistemin kullanımında diğer menü'deki fonksiyonlar ve sisteme etkileri şöyle sıralanabilir:

Diğer Menü'ye "ÇIK" tuşuna basılarak geçildiği daha önce belirtilmiştir.

İkinci menüde;

F1 tuşu ile kursör EĞİM SIFIRLAMA satırına gelir. YAP/EVET tuşuna basılarak aracın yer düzlemindeki X ve Y eğim değerleri hafızaya alınır. Bu değerler namlu hareketi sırasında veya seri atışlar sırasında değişebilir. Meydana gelen değişim fark miktarı, hedef koordinat değeri ( ATIŞ ) üzerine olan etkisi hesaplanarak otomatik olarak eklenir veya çıkartılır ve bu hesaplanan değer MY. D. A. satırına komut verildiği anda yazılır. Namlu açısı otomatik olarak bu noktaya gelecektir. Ancak namlunun deniz seviyesine göre mutlak açısı ATIŞ satırına yazılan değerdir. Araç mevzi değiştirdiğinde mutlaka Eğim Sıfırlama işlemi yapılmalıdır.

F2 tuşu işlevi daha önce tanımlanmıştır.

F3 tuşu ile kursör ELLE KUMANDA satırına gelir. YAP/EVET tuşuna basıldıktan sonra sistem otomatik olarak kontrol ekranı düzenine döner ve alt kısma "Oklara Bas Çalıştır" mesajı gelir. Ekranda bulunan değerler sistemin daha önce kaldığı en son değerlerdir. Sağ-sol okları ile kule, yukarı-aşağı okları ile namlu okların basılı tutulduğu süre kadar hareket eder ve denge basıncıda otomatik kontrol edilir. NAMLU AÇISI değeri de hareket miktarı kadar değişimi gösterir. Elle kumanda modundan çıkmak için bu moda girmek için yapılan işlem tekrar edilir ve sistem otomatik olarak kontrol ekranı düzenine döndüğünde "Oklara Bas Çalıştır" yazısı ekrandan silinmiştir.

F4 tuşu ile kursör SİSTEM TEST satırına gelir. YAP/EVET tuşuna basıldıktan sonra KONTROL KUTUSU ve KONSOL SELFTEST işlemine tabi tutulur ve herhangi bir problem yok ise TAMAM mesajı ile bu moddan otomatik olarak çıkılır. Eğer sistem birimlerinden herhangi bir şekilde şüphe duyulduğunda bu işlem yapılmalıdır. Eğer test doğru tamamlanmadıysa arızayı yazılı olarak belirtir.

F5 tuşu ile kursör ATIŞ BİLGİ satırına gelir. YAP/EVET tuşuna basıldıktan sonra ekran Şekil 5.6'daki düzeni alır.

**** ATIS BİLGİ ****					
F1	NO	:	0000		
F2	TAPA	:	0000		
	MERMİ	:	0000		
F3	HIZI	:	0000		
			Tipi		Adet
F4	MERMİ	:	000		0000
			Hakkı		Kese
F5	BARUT	:	0000		0000
			Gün	Ay	yıl
F6	TARİH	:			199

Çıkış İçin <CIK>

Şekil 5.6. Sistemin atış bilgisi bölümü konsol ekran görüntüsü

F6 tuşu ile kursör TERCİHLER satırına gelir. YAP/EVET tuşuna basıldıktan sonra ekranın alt kısmında EĞİM GÖSTERİM ? (E/H) mesajı çıkar. Eğer kulenin deniz seviyesine göre eğim ölçer tarafından ölçülen değeri ve bu değerlerin değişimi gözlenmek isteniyorsa YAP/EVET tuşuna, eğer istenmiyorsa DUR/HAYIR tuşuna basınız. Bu soruya verilen cevabın ardından ekranın aynı yerinde BASINÇ GÖSTERİM? (E/H) mesajı çıkar.

Eğer namlunun denge basıncı değeri görmek isteniyorsa YAP/EVET, eğer görmek istenmiyorsa DUR/HAYIR tuşuna basınız. Bundan sonra ekran aynı düzenini koruyacaktır. Ancak EĞİM GÖSTERİM ? sorusuna EVET, BASINÇ GÖSTERİM? 'e HAYIR cevabı verildiyse, kontrol ekranı Şekil 5.7'deki düzeni alacaktır.

BAK52		Yan	Yuks
F1 ATIS	:	0000	0000
F2 ATMD	:	0000	0000
F3 NAMLU			
ACISI	:	3200	0000
MY. D. A.	:		
-----			
KULE		X	Y
EGİM	:	0012	0015

Diğer Menü İçin <CIK>

Şekil 5.7. Sistemin kule eğim gösterim konsol ekran görüntüsü

EGİM GOSTERİM? sorusuna HAYIR, BASINC GOSTERİM? sorusuna EVET cevabı verilseydi ekran Şekil 5.8'deki gibi olacaktır.

<b>BAK52</b>	<b>yan</b>	<b>Yuks</b>
<b>F1 ATIS</b>	<b>: 0000</b>	<b>0000</b>
<b>F2 ATMD</b>	<b>: 0000</b>	<b>0000</b>
<b>F3 NAMLU</b>		
<b>ACISI</b>	<b>: 3200</b>	<b>0000</b>
<b>MY. D. A.</b>	<b>:</b>	
-----		
<b>DENGE</b>		
<b>BASINC</b>	<b>: 0092</b>	<b>Bar</b>
<b>Diger Menu Icin &lt;CIK&gt;</b>		

Şekil 5.8. Sistemin denge basınç gösterim konsol ekran görüntüsü

Eğer her iki soruyada EVET cevabı verilseydi ekran Şekil 5.9'daki gibi olacaktır.

<b>BAK52</b>	<b>yan</b>	<b>Yuks</b>
<b>F1 ATIS</b>	<b>: 0000</b>	<b>0000</b>
<b>F2 ATMD</b>	<b>: 0000</b>	<b>0000</b>
<b>F3 NAMLU</b>		
<b>ACISI</b>	<b>: 3200</b>	<b>0000</b>
<b>MY. D. A.</b>	<b>:</b>	
-----		
<b>DENGE</b>		
<b>BASINC</b>	<b>: 0092</b>	<b>Bar</b>
<b>KULE</b>	<b>X</b>	<b>Y</b>
<b>EGİM</b>	<b>0012</b>	<b>0015</b>
<b>Diger Menu Icin &lt;CIK&gt;</b>		

Şekil 5.9. Sistemin denge basınç ve kule eğim gösterim konsol ekran görüntüsü

F7 tuşu ile kursör EL ÇARKI KNT. ( El Çarkı ile Kontrol ) satırına gelir. YAP/EVET tuşuna basıldıktan sonra ekran otomatik olarak diğer menüye döner ve ekranın alt kısmına "Oklara Bas Çalıştır" mesajını yazar. Bu durumda namılu el çarkı ile yukarı hareket ettirmek istiyorsanız yukarı ok tuşuna , aşağı hareket ettirmek aşağı ok tuşuna basınız ve el çarkını çeviriniz. Sistem denge basıncını otomatik kontrol ettiği için

el çarkı ile zorlanmadan namlu kaldırılacak veya indirilecektir. Eğer yön değiştirilecekse mutlaka önce istakamet yönünü belirten ok tuşuna basılmalıdır. El Çarkı ile Kumanda modundan çıkmak için bu moda girmek için yapılan işlem tekrar edilir ve sistem otomatik olarak kontrol ekranı düzenine döndüğünde "Oklara Bas Çalıştır" yazısı ekrandan silinmiştir.

Bu menüde görülen F1'den F7'ya kadar olan kontrollerin ve fonksiyonların birkaç tanesi aynı anda kullanılabilir.

### 5.5. Sistemin Baiks Bağlantısı

BAK52 sistemine BAIKS bağlantısı için Baiks terminalinden gelen bilgi taşıma kablosu ucundaki 15 Pin Dişi soketi Konsol Kutusu altındaki Baiks olarak isimli 15 Pin erkek sokete takınız ve sabitleme vidalarını sıkınız. Bu bağlantı yapıldıktan Baiks'den mesaj alınmak istendiğinde F1 tuşuna basıp bekleyiniz. Baiks terminalinden mesaj geldiğinde sistem otomatik olarak gelen değerleri Yan ve Yükseliş hanelerine yazar ve ekranın alt kısmında BAIKS MESAJ ALINDI yazılı mesajını kısa bir süre basar. İşlemin gereğini yapmak için operatörden YAP/EVET tuşuna basılmasını bekler. YAP/EVET tuşuna basıldığında sistem Eğitim değişimleri ve ATMD değerlerini de kontrol ederek kuleyi ve namluyu istenen koordinatlara getirir.

Baiks bağlantısının yapılmış olması manuel kumanda yapmayı engellemez.

Baiks'den iletişim sırasında bozulmaya uğramış mesaj gelirse , değerlendirmeye almaz.

Örnek:

F1 tuşuna basınız ve kursörün F1 ATIŞ satırına geldiğini görünüz.

Baiks'den telsiz kanalı ile gönderilen ve Baiks terminali ile sisteme ulaşan bilgi;

Yan : 3250

Yüks : 0645

Mesaj sisteme ulaşp YAP/EVET basılmadan önceki ekran durumu Şekil 5.10'daki gibidir.

BAIKS MESAJ ALINDI yazısı kısa bir süre sonra kaybolacaktır.

<b>BAK52</b>	<b>yan</b>	<b>yuks</b>
<b>F1 ATIS</b>	<b>: 3250</b>	<b>  0645</b>
<b>F2 ATMD</b>	<b>: 0000</b>	<b>  0000</b>
<b>F3 NAMLU</b>		
<b>ACISI</b>	<b>: 3200</b>	<b>  0000</b>
<b>MY. D. A.</b>	<b>:</b>	<b> </b>
-----		
<b>BAIKS MESAJ ALINDI</b>		
<b>Diger Menu Icin &lt;CIK&gt;</b>		

Şekil 5.10. Sistem baiks'den mesaj aldığındaki konsol ekran görüntüsü

## 5.6. Sistemin Bakımı

BAK52 sisteminde mekanik birimlerin (Motor redüktörleri, enkoder kupları noktaları, basınç ölçer hidrolik bağlantısı) standart usüllerle yapılacak bakımlarının dışında özel bakım talimatlarına ihtiyacı yoktur. Ancak eğim ölçer ve basınç ölçer birimlerinin 2 yılda 1 kez kalibrasyon kontrollerinin yapılması gerekir. Bunun için;

### 5.6.1. Eğim ölçer kalibrasyonu

Bu test ancak 5. kademe bakım-onarım atölyelerinde yapılacaktır. '+100' milyem eğimde 8 V, '-100' milyem eğimde +4 V verecek biçimde önce '0' milyemde +6 V ayarı daha sonra diğer eğimlerde kazanç ayarları yapılır.

### 5.6.2. Basınç ölçer kalibrasyonu

Bu test ancak 5. kademe bakım onarım atölyelerinde yapılabilir. Elektronik basınç ölçer hidrolik sisteme bağlanır. '80' bar basınç verilir. Çıkış geriliminin 6V olması potansiyometreler ile ayarlanır. '50' bar basınçta 4V çıkış vermesi kontrol edilir.

## 5.7. Sistemin Onarımı ve Arıza Tespiti

BAK52 sistemi, arıza yapma ihtimali olan bölümlerini sürekli kontrol altında tutar ve herhangi bir arıza durumunda bunu operatöre konsol ekranında yazılı mesaj olarak gösterir. Bu mesajların sebepleri ve yapılması gereken müdahaleler şunlardır.

**KONT. KUTUSU CEVAP YOK :** Konsolun tam faal olduğunu, arızanın kontrol kutusu bilgisayarından kaynaklandığını veya aradaki bağlantı kablosundan kaynaklandığını gösterir. Önce bağlantı kablosunun sağlığını kontrol ediniz. Arıza giderilmemişse kontrol kutusu bilgisayar kartını değiştirin.

**İLETİŞİM HATA :** Bu hata sistemler arasında iletişimin var olduğunu ancak mesajların bozuk geldiğini gösterir. Bilgisayarlar arası iletişim hatası konsol bilgisayarından veya kontrol kutusu bilgisayarından veya aradaki bağlantı kablosundan kaynaklandığını gösterir. Önce bağlantı kablosunun sağlığını kontrol ediniz. Arıza giderilmemişse sıra ile önce kontrol kutusu bilgisayarının kartını, eğer düzelmezse konsol bilgisayar kartını değiştiriniz.

**KNSL BELLEK ARIZASI :** Konsol bilgisayarının bellek arızasını belirtir. Konsol bilgisayar kartını değiştiriniz.

**MATEMATİK HATASI :** Konsol bilgisayar kartını değiştiriniz.

**KON. BELLEK ARIZASI :** Kontrol kutusu bilgisayar sisteminin bellek arızasını belirtir. Kontrol kutusu bilgisayar kartını değiştiriniz.

**YAN MOTOR AŞIRI AKIM** : Yan motor mekanik aksamında bir sıkışma meydana gelmiş olabilir. Varsa gideriniz veya Yan motor sürücü kartını değiştiriniz.

**YÜKS MOTOR AŞIRI AKIM** : Yükseliş motor mekanik aksamında bir sıkışma meydana gelmiş olabilir. Aynı zamanda denge basıncının açığa göre düzgün ayarlanıp ayarlanmadığını kontrol ediniz. Aksaklık varsa gideriniz veya yükseliş motor sürücü kartını değiştiriniz.

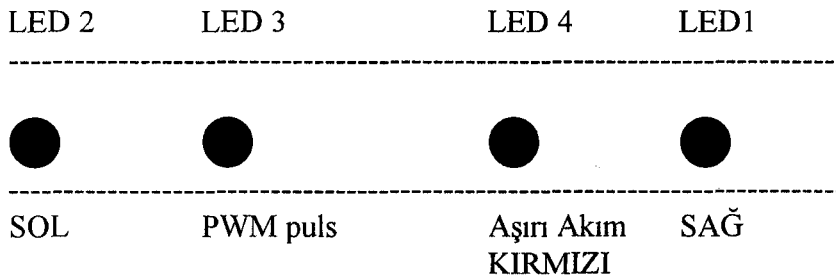
**EĞİM ÖLÇER ARIZA** : Eğim ölçer' in kalibrasyonu kontrol ediniz veya değiştiriniz. Arıza devam ediyorsa Ölçüm ve Kontrol kartını değiştiriniz.

**YAĞ BASINCI ARIZA** : Basınç ölçer'in kalibrasyonunu kontrol ediniz veya değiştiriniz. Arıza devam ediyorsa Ölçüm ve Kontrol kartını değiştiriniz.

**BESLEMEDE ARIZA** : Sistemin iç besleme gerilim regülatörlerindeki arıza belirtimidir. Besleme sistemini kontrol ediniz.

**YRD MTR. ÇALIŞTIR** : Bu ikaz ana akü geriliminin 23 V altına düştüğü hallerde ekrana gelir. Topun yardımcı motoru 30 dk. süre ile çalıştırılırsa aküler şarj edilir ve bu mesaj kalkar.

**KONTROL KUTUSU IŞIKLI LEDLER YARDIMI İLE ARIZANIN SAPTANMASI**: Kontrol kutusu içersindeki motor sürücü kartları çıkışlarına monte edilmiş ışıklı ledlere bakılarak arıza saptanabilir.



Şekil 5.11. Sistem kontrol kutusu üzerindeki ledlerin diziliş görüntüsü

Konsoldan hareket komutu verildiğinde hareket yönüne uygun olarak SAĞ (LED1) veya SOL (LED 2) ve PWM puls (LED3) beraber yanarak hareket etmelidir.

Bir mekanik sıkışma halinde aşırı akım LED4 parlayacaktır.

## 6. SONUÇLAR, YORUM ve ÖNERİLER

Bu tez çalışmasında, tankların atış sistemi için; bilgisayarlı atış kontrol sistemi geliştirilmiştir. Bilgisayarlı atış kontrol sistemi tank ve obüslerinin namlu açısını dikey (yükseliş) ve yatay (yan) olarak istenen koordinatlara ayarlanmasını, bilgisayar kontrolü ile sağlayan bir sistemdir.

Bu sistemde kullanılan, namlunun hareketini sağlayan yan ve yükseliş adı verilen dc servo motorların transfer fonksiyonunu bulmak için; arabirimler tasarlanmıştır. Tasarlanan arabirimlerin devreleri Şekil 4.2, 4.3, 4.4'te verilmiştir. Şekil 4.2'de sisteme ait servo motorların kontrolünü sağlayan CPU devresi tasarlanmıştır. Şekil 4.3'te servo motorlara bağlı olan enkoderlerin değerlerini okuyan ve bu değerleri kontrol devresi olan CPU'ya ileten enkoder sayım devresi tasarlanmıştır. Şekil 4.4'de sistemin kontrol devresi olan CPU'dan gönderilen sinyallerin işlenip motora iletilmesini sağlayan, motor sürücü devresi tasarlanmıştır. Ayrıca Şekil 4.2'de tasarlanmış olan CPU devresinin işletilmesini sağlayan kontrol programı Ek-1'de verilmiştir.

Yazılmış olan kontrol programını desteklemek için de arabirimler gerçekleştirilmiştir. Tasarlanan arabirimlerin devreleri Şekil 4.6, 4.7, 4.8'te verilmiştir. Şekil 4.6'de sisteme ait servo motorların kontrolünü sağlayan CPU devresi tasarlanmıştır. Şekil 4.7'te servo motorlara bağlı olan enkoderlerin değerlerini okuyan ve bu değerleri kontrol devresi olan CPU'ya ileten enkoder sayım devresi tasarlanmıştır. Şekil 4.8'de sistemin kontrol devresi olan CPU'dan gönderilen sinyallerin işlenip motora iletilmesini sağlayan, motor sürücü devresi tasarlanmıştır. Ayrıca Şekil 4.6'de tasarlanmış olan CPU devresinin işletilmesini ve sisteme ait kontrol mekanizmasının gerçekleştirilmesini sağlayan kontrol programı Ek 2'de verilmiştir.

Bölüm ikide sistem için yazılan kontrol programına ait, kontrol denklemlerinin çıkarılması verilmiştir. Kontrol programı yazılırken, bu denklemlerde elde edilen ifade ve katsayılar göre düzenlenmiştir.

Bölüm üçte sistemde kullanılan mikrokontrolörün, servo motor ve enkoderin iç yapısı özellikleri kısaca özetlenmiştir. Mikrokontrolörün kolay programlanabilmesi için gerekli bazı özelliklerine değinilmiştir.

Bölüm dörtte; ilk olarak sistemde kullanılan servo motorun hareket eğrisinin, daha iyi incelenebilmesini sağlamak için yazılmış programın ve bu programa ait akış diyagramının açıklaması verilmiştir. Daha sonra ana sistemdeki, servo motorların ve diğer ünitelerin kontrol edildiği, atış kontrol programının ve bu programa ait akış diyagramının açıklamaları verilmiştir.

Bölüm beşte; tasarlanmış olduğumuz sistemin, hangi ana bölümlerden oluştuğu ve bölümlerin kısa açıklamaları verilmiştir. Ayrıca tasarlanmış sistemin işletilmesi hakkında görsel ve açıklamaya dayalı bilgiler verilmiştir.

Tasarlanmış olan sistem oldukça karmaşık bir yapıya sahiptir. Bu yüzden sistem içinde dc servo motor haricinde; hidrolik sistem, sistemdeki değerlerin görüntülediği ekran ve tuş takımı, bu sistemleri kontrol eden konsol ünitesi (mikroişlemcili devre), baiks haberleşmesi, basınç ve eğim ölçer modül sistemlerinden detaylı bir şekilde bahsedilmemiştir. Diğer bölümler ayrı ayrı tez konuları olarak düşünülebilir.

Bu tez çalışmasında işlenen konulara ek olarak, projenin tamamına öneriler getirmek daha uygun olacaktır. Bu yolla projenin daha sağlıklı ve güvenli çalışması mümkün olacaktır:

- Sisteme ait olan eğim ölçer ve basınç ölçer özel tasarım olarak gerçekleştirilmiştir. Bu modüller istenen performansı sağlayamamıştır. Bu sebepten dolayı modülleri daha hassas hale getirmek için bir çalışma gerçekleştirilebilir.

- Eğim ölçer ve basınç ölçerler modüllerinin hassas gerçeklenmiş modellerini yurtdışından temin edilmesi ve bu yeni modelleri, kendi arasında karşılaştırma yaparak performansı en iyi olan modülü belirlemek için bir çalışma gerçekleştirilebilir.

- Sisteme eğim ölçer modülü yerine, gyro modülü takılabilmesi için gerekli düzenleme ve böyle bir modül değişiminin sisteme kazandırdıkları ve kaybettirdikleri ile ilgili bir çalışma gerçekleştirilebilir.

- Sistem için yazılan kontrol programını daha kısa hale getirmek ve sistemin işleyiş performansını artırıcı çalışma gerçekleştirilebilir.

- Tank ve obüs içine yerleştirilecek olan sistem modülleri daha dayanıklı, az yer kaplayan ve montajı kolay hale getirilebilir.

Yapılması istenen öneriler yapılmadığı şartlar altında bile, tasarlanmış olan sistem oldukça düzgün, güvenli ve istenir şekilde Kara Kuvvetleri Komutanlığına ait fabrikalarda ve tanklarda çalışmasına devam etmektedir.

## KAYNAKLAR

1. 80C51 Based 8 Bit Microcontrollers Data Handbook, Philips Semiconductors, 1994
2. The European Cmos Selection, Motorola Inc., 1985
3. Optoelectronics Designer's Catalog, Hewlett Packard, 1991-1992
4. Omron Servomotor Seminer Notları, Omron Inc., 1997
5. Computer Control Of Manufacturing System, Mcgraw-Hill Inc., 1983
6. Design And Analysis Of A Modular CNC System, Computers In Industry, 1990
7. WILLIAM J. Palm, Control System Engineering, John Wiley & Sons Inc., USA, 1986
8. Extended Basic-51 Cross-Compiler ( BXC51), Binary Technology Inc., 1989
9. OGATA Katsuhiko, Modern Control Engineering, Prentice-Hall Inter Inc., 1990

## EK-1

```

REM *****
REM ***** SERVO MOTORUN TRANSFER FONKSİYONUNU *****
REM ***** BULMA PROGRAMI *****
REM *****
REM ***** CUMHUR AKIN *****
REM ***** YÜKSEK LİSANS BİTİRME TEZİ *****
REM *****

```

```

5 A%= 0A000H: B%=0C000H: C%=08000H
10 GOSUB 900

```

```

12 PRINT "*****"
13 PRINT "***** CUMHUR AKIN *****"
15 PRINT "***** YUKSEK LISANS BITIRME TEZI *****"
16 PRINT "***** MOTOR KONTROL PROGRAMI *****"
18 PRINT "*****"
19 PRINT " 1  MOTORU CALISTIRMA, DEGER SAKLAMA VE DURMA "
20 PRINT " 2  SUREKLI SAYICIYI OKUMA "
21 PRINT " 3  SAKLANAN DEGERLERI OKUMA "
22 PRINT " 4  SISTEM TEST"
23 PRINT " 9  SISTEMI VE SAYICIYI SIFIRLAMA(-32768)"

```

```

REM ANA LOOP

```

```

50 GR%=GET :REM dis komut bak
55 IF (GR%>47 .AND. GR%<58) THEN GOSUB 190
60 GOTO 50

```

```

REM *** klavyeden gelen değere bak ve verilen emri yap

```

```

REM "0"=48 ile "9"=57
190 EMR%=GR%-48:REM      0  1  2  3  4  5  6  7  8  9
REM                      0  1  2  3  4  5  6  7  8  9
192 ON EMR% GOSUB      196,200,410,300,800,196,19€,196,196,900
196 GOTO 20000
200 PRINT "MOTOR HAREKETE BASLADI"
$asm
CALL MEMSİL
CALL MOTOR
$basic
220 PRINT "MOTORUN HAREKETİ DURDURULDU"
224 GOTO 20000

```

```

300 PRINT "ALINAN DEGERLER":SAYAC%=0:ADR%=1000:PRINT "ADR=",ADR%
304 SAYAC%=SAYAC%+1:IF SAYAC%>200 THEN SAYAC%=1:GOTO 310
306 SAYI%=XBY(ADR%)+XBY(ADR%+1)*256:PRINT SAYI%,
307 ADR%=ADR%+1:IF ADR%>7000H THEN PRINT "ISLEM BITTI":GOTO 20000
308 GOTO 304
310 PRINT:PRINT "DEVAM ICIN HERHANGI BIR TUSA,ISLEMI BITIRMEK ICIN 9'A BAS"
312 GR#=GET:IF GR#=57 THEN PRINT "OKUMA BITTI":GOTO 20000

```

```

314 IF GR#=0 THEN GOTO 312
317 PRINT "ADR=",ADR%:GOTO 306

```

```

REM ***** YAN DEGERI OKU VE EKRANDA GOSTER *****

```

```

$asm

```

```

    CALL SAYOKU

```

```

$basic

```

```

446 YAN%=YANL#+YANO##256:PRINT YAN%,

```

```

448 GR#=GET:IF GR#=57 THEN GOTO 20000 ELSE GOTO 410:REM 9'A BASILIRSA CIK

```

```

800 PRINT "SISTEM TEST"

```

```

820 GOTO 20000

```

```

900 PRINT "SIFIRLAMA"

```

```

$asm

```

```

    CALL YANSIFIRLA

```

```

    SETB P1.4

```

```

$basic

```

```

920 GOTO 20000

```

```

1000 REM

```

```

$asm

```

```

MEMBAS    EQU    1000H

```

```

MEMSON    EQU    7000H

```

```

MEMORT    EQU    (MEMSON-MEMBAS)/2

```

```

MOTOR:

```

```

        CLR P1.4                ;MOTOR CALISTIRILDI

```

```

        MOV DPTR,#MEMBAS

```

```

        MOV R3,DPL

```

```

        MOV R4,DPH

```

```

M1:    CALL OKU
        MOV A,DPH
        JB RI,CIK
        CJNE A,#HIGH(MEMORT),M1

```

```

        SETB P1.4                ;MOTOR DURDURULDU

```

```

DEACCEL:

```

```

        MOV DPTR,#MEMORT

```

```

        MOV R3,DPL

```

```

        MOV R4,DPH

```

```

M2:    CALL OKU
        MOV A,DPH
        JB RI,CIK
        CJNE A,#HIGH(MEMSON),M2

```

```

CIK:   MOV A,SBUF
        CLR RI

```



## EK-2

```

REM ***** SİSTEM KONTROL PROGRAMI *****
REM *****
REM ***** CUMHUR AKIN *****
REM ***** YÜKSEK LİSANS BİTİRME TEZİ *****
REM*****

```

```
5 PORT1 = 0:KP#=255:KI#=255:KEGIM=1.9608:KEGIMC=-300
```

```
7 PK=1.5686:PC=-160:
```

```
10 A% = 0A000H: B%=0C000H: C%=08000H:T%=100
```

```
11 GOSUB 9998: REM sistemi sıfırla
```

```
    REM assembler degiskenlerini set et
```

```
15 YANOF% = 0:YUKOF% = 0:TURSA# = 30:GOSUB 445
```

```
17 SAR# = 0:ELLEM# = 0:YYH# = 0:YYL# = 0:YYO# = 0
```

```
18 YANPWM# = 0:YUKPWM# = 0:HATAL# = 0:HATAO# = 0:HATAH# = 0
```

```
19 EYANHATAL# = 0:EYANHATAO# = 0:EYANHATAH# = 0
```

```
20 EYUKHATAL# = 0:EYUKHATAO# = 0:EYUKHATAH# = 0
```

```
21 V# = 0:GR2% = 0:MERMIVAR# = 0:CIKBSNCADR# = 0:INBSNCADR# = 0
```

```
22 TEMP1H# = 0:TEMP1O# = 0:TEMP1L# = 0:TEMP2H# = 0:TEMP2O# = 0:TEMP2L# = 0
```

```
23 YUKSONL# = 0:YUKSONO# = 0:YUKSONH# = 0:YANSONL# = 0:YANSONO# = 0:YANSONH# = 0
```

```
    REM MERMI BOS YUKSELIS 1B00 2*3600/6400=1.125=>14,26 milyem bir basinc sayisi
```

```
30 PMEM% = 1B00H:GOSUB 165
```

```

31 DATA 103,103,103,102,101,100,100 :REM (-90)-0 milyem
32 DATA 100,99,97,97,96,95,94,93 :REM 0-99 milyem
33 DATA 92,92,91,90,89,89,88 :REM 114-199 milyem
34 DATA 87,86,85,84,83,83,82 :REM 214-299 milyem
35 DATA 81,80,79,79,78,78,77 :REM 313-399 milyem
36 DATA 77,76,75,75,74,74,73 :REM 413.54-499.1 milyem
37 DATA 73,72,71,71,70,69,68 :REM 513-598.92 milyem
38 DATA 67,66,65,64,63,62,61 :REM 613.18-698.74 milyem
39 DATA 60,60,59,58,57,56,55 :REM 713-798.56 milyem
40 DATA 55,54,53,53,52,51,50 :REM 812.82-898.38 milyem
41 DATA 50,50,49,49,48,48,48 :REM 912.64-998.2 milyem
42 DATA 47,47,46,46,44,42,41 :REM 1012.46-1098.02 milyem
43 DATA 40,40,40,40 :REM 1112.28-1155.6 milyem

```

```
    REM MERMI DOLU YUKSELIS 1C00
```

```
50 PMEM% = 1C00H:GOSUB 165
```

```

51 DATA 98,98,97,96,95,94,93 :REM (-90)-0 milyem
52 DATA 93,92,91,90,89,88,88,87 :REM 0-99 milyem
53 DATA 86,86,85,84,84,83,83 :REM 114-199 milyem
54 DATA 83,82,82,81,80,79,78 :REM 214-299 milyem
55 DATA 77,76,76,75,75,74,73 :REM 313-399 milyem
56 DATA 72,72,71,71,70,70,69 :REM 413.54-499.1 milyem
57 DATA 69,68,67,66,66,65,65 :REM 513-598.92 milyem
58 DATA 64,64,63,62,62,61,60 :REM 613.18-698.74 milyem
59 DATA 59,59,58,57,56,55,54 :REM 713-798.56 milyem
60 DATA 54,53,52,52,51,50,49 :REM 812.82-898.38 milyem
61 DATA 49,49,48,48,47,47,47 :REM 912.64-998.2 milyem
62 DATA 46,46,45,45,43,41,40 :REM 1012.46-1098.02 milyem

```

```

63 DATA      38,38,37,36                :REM 1112.28-1155.6 milyem
      REM MERMI BOS ALCALIS ID00
70 PMEM%=1D00H:GOSUB 165
      71 DATA      96,96,96,95,94,93,93          :REM -90-0 milyem
      72 DATA 93,92,91,90,89,88,87,86          :REM 0-99 milyem
      73 DATA 85,84,83,82,81,80,80            :REM 114-199 milyem
      74 DATA 79,79,78,78,77,76,75            :REM 214-299 milyem
      75 DATA 75,74,73,72,71,70,69            :REM 313-399 milyem
      76 DATA 68,67,66,66,65,65,64            :REM 413-499 milyem
      77 DATA 64,63,63,62,62,61,61            :REM 513-598 milyem
      78 DATA 60,60,59,59,58,57,57            :REM 613-698 milyem
      79 DATA 56,55,54,54,53,53,52            :REM 713-798 milyem
      80 DATA 52,51,51,50,49,48,47            :REM 812-898 milyem
      81 DATA 46,46,45,45,44,44,43            :REM 912-998 milyem
      82 DATA 42,41,40,39,38,37,36            :REM 990-1098 milyem
      83 DATA 35,34,33,32                    :REM 1112-1155 milyem

```

```

      REM MERMI DOLU ALCALIS 1E00
90 PMEM%=1E00H:GOSUB 165
      91 DATA      91,91,90,90,89,89,88          :REM -90-0 milyem
      92 DATA 87,86,89,88,87,86,85,84          :REM 0-99 milyem
      93 DATA 83,82,81,80,79,78,78            :REM 114-199 milyem
      94 DATA 77,77,76,76,75,75,74            :REM 214-299 milyem
      95 DATA 73,72,72,71,70,69,69            :REM 313-399 milyem
      96 DATA 69,68,68,67,66,66,65            :REM 413-499 milyem
      97 DATA 64,64,63,62,62,61,60            :REM 513-598 milyem
      98 DATA 59,58,57,57,56,56,55            :REM 613-698 milyem
      99 DATA 54,54,53,53,52,51,50            :REM 713-798 milyem
      100 DATA 49,48,47,47,46,46,45            :REM 812-898 milyem
      101 DATA 44,43,42,41,40,39,38            :REM 912-998 milyem
      102 DATA 37,36,35,34,33,32,31            :REM 1012-990 milyem
      103 DATA 30,29,29,29                    :REM 1112-1155 milyem

```

```
$ASM
```

```

      CALL YANSIFIRLA
      CALL YUKSIFIRLA

```

```
$BASIC
```

```
130 CIKBSNCADR#=1BH:INBSNCADR#=1DH :REM mermi yuksuz basinc egrisi
```

```
132 PRINT "*****"
```

```
133 PRINT "***** CUMHUR AKIN *****"
```

```
135 PRINT "***** YÜKSEK LİSANS BİTİRME TEZİ *****"
```

```
136 PRINT "***** SİSTEM KONTROL PROGRAMI *****"
```

```
138 PRINT "*****"
```

```
      REM ANA LOOP
```

```
140 GR%=GET                                :REM dis komut bak
```

```
142 IF (GR%>47 .AND. GR%<68) THEN GR1%=GR% ELSE GOTO 148
```

```
144 IF GR1%=GR2% THEN GOSUB 190:GR2%=0 ELSE GR2%=GR1%
```

```
148 IF (EMR%>14) .OR. (CARK%=1) THEN GOTO 160
```

```
149 GOSUB 9998
```

```
150 GOTO 140
```

```
160 IF (EMR%<17).AND.(CARK%=0) THEN GOTO 140 :REM yan ilerleme icin
```

```
$ASM
```

```

; yukselis hareket icin
TUR          EQU    30
AYARLA:     CALL  ELLEPAYAR ;basinc istenen seviyede ise ACC 0 olur
           JNZ   CIK

           MOV   DPTR,#IV_CARK
           CALL  IGETVAR
           CJNE  R0,#0,CIK    ;CARK%=1 ise motoru calistirmadan cik
                               ;basinc ayarlandi ise

           MOV   P1,BV_ELLEM ;motoru calistirmek icin
           CALL  KSBK        ;motorlar kısa devre bak

CIK:        JMP   LN140

$BASIC
165 FOR JYUK%=0 TO 6:ADR%=INT(PMEM%-6+JYUK%+0FFH):GOSUB 168:NEXT
JYUK%:REM - milyemler
166 FOR JYUK%=0 TO 81:ADR%=INT(PMEM%+JYUK%):GOSUB 168:NEXT JYUK%:REM +
milyemler
167 RETURN
168 READ X:PADC SAYI=(X-PC)/PK:XBY(ADR%)=LOW(INT(PADC SAYI)):RETURN

           REM *** konsolu dinle ve verilen emri yap
REM "0"=48 ile "9"=57
190 EMR%=GR%-48:REM      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
REM      0 1 2 3 4 5 6 7 8 9 : ; < = ? @ A B C
192 ON EMR% GOSUB
200,800,196,196,196,196,240,196,225,300,310,500,502,196,360,350,370,380,400
196 RETURN

           REM degerleri oku ve konsola bildir
200 SAR#=0:GOSUB 410:GOSUB 270
$ASM
           CALL DGRYOLLA
$BASIC
218 BASINCBAR% =INT(GP##PK+PC)
219 PH0.YANO#,YANL#," ",YUKO#,YUKL#," ", " GP=",GP#,:PRINT BASINCBAR%,"
":PH0.SAR#,
220 PH0." ",YANOFs%,YUKOFs%," CARK=",CARK%
222 RETURN

           REM "9" gelince sistem reset
225 CARK%=0:GOSUB 9998:RETURN

           REM sistem test
240 SAR#=0
242 FOR ADR% = 5 TO 7: GOSUB 460: REM 5>0V 6>5V 7>VREF 5V
243 ON ADR%-5 GOSUB 258,260,260
244 NEXT ADR%
246 GOSUB 270 : REM aku gerilimini oku
248 GOSUB 420 : REM EGX EGY ve GP oku
$ASM
           CALL HFZTST
           MOV  A,BV_SAR
           CALL YOLLA
$BASIC

```

252 RETURN

258 IF DG%>10 THEN SAR#=3  
 259 GOTO 252  
 260 IF DG%<240 THEN SAR#=3  
 261 GOTO 252

REM aku gerilimini oku

270 ADR%=3:GOSUB 460:IF DG%<122 THEN SAR#=8:REM <22.5V aku gerilimi yrd mtr calistir

REM egim olcer ariza

272 IF (ABS(EGX%)>220).OR.(ABS(EGY%)>220) THEN SAR#=4  
 274 IF GP#>204.OR. GP#<60 THEN SAR#=5 : REM basinc olcer ariza  
 276 RETURN

300 REM yan sayici sifirla

\$ASM

CALL YANSIFIRLA

\$BASIC

301 RETURN

310 REM yuks sayici sifirla

311 SAR#=0

\$ASM

CALL YUKSIFIRLA ;yuk sayiciyi sifirla

CALL KONSOLDANDGROKU ;varsa konsoldan ofset degerlerini oku

\$BASIC

314 IF SAR#<>0 THEN RETURN

316 YANOF%=(TEMP10#\*256+TEMP1L#):YUKOFS%=(TEMP20#\*256+TEMP2L#

324 RETURN

REM elle kumanda

350 M#=10H:PORT1=M#:GOTO 390: REM yan motor saat yonunde  
 360 M#=20H:PORT1=M#:GOTO 390: REM yan motor saat yonu aksi  
 370 M#=0:ELLEM#=40H:GOTO 390: REM yuks motor saat yonu YUKSELIR 17 A  
 380 M#=0:ELLEM#=80H:GOTO 390: REM yuks motor saat yonu ALCALIR 18 B

REM TEMP20 ve TEMP10 kısa devre motor kont icin kullaniliyor

390 V#=0:TEMP20#=0:TEMP10#=0:TURSAY#=1

391 IF EMR%=17 THEN YON#=1 ELSE YON#=0 :REM basinc icin

392 RETURN

REM cark kumanda

400 CARK%=1:RETURN

REM \*\*\*\* YAN% YUK% oku

410 GOSUB 445

416 YANO#=#HIGH(YAN%):YANL#=#LOW(YAN%):YUKO#=#HIGH(YUK%):YUKL#=#LOW(YUK%)

REM \*\*\*\* GP# EGX% EGY% oku \*\*\*\*\*

420 FOR ADR% = 0 TO 2: GOSUB 460

425 ON ADR% GOSUB 430, 431, 432

428 NEXT ADR%

429 RETURN

430 GP# = LOW(DG%): RETURN

431 GOSUB 440:EGX% = INT(S3):RETURN:REM milyem



```

936 REM
$ASM
    CALL  YANSAYOKU
; YANSON% = YAN% + DYAN%   sayim cinsinden
    MOV   R0,BV_YANL
    MOV   R2,BV_YANO
    MOV   R4,#0
    MOV   R1,BV_TEMP1L
    MOV   R3,BV_TEMP1O
    MOV   R5,BV_TEMP1H
    CALL  IADD3
    MOV   BV_YANSONL,R1
    MOV   BV_YANSONO,R3
    MOV   BV_YANSONH,R5

    CALL  YUKSAYOKU
; YUKSON% = YUK% + DYUK%
    MOV   R0,BV_YUKL
    MOV   R2,BV_YUKO
    MOV   R4,#0
    MOV   R1,BV_TEMP2L
    MOV   R3,BV_TEMP2O
    MOV   R5,BV_TEMP2H
    CALL  IADD3
    MOV   BV_YUKSONL,R1
    MOV   BV_YUKSONO,R3
    MOV   BV_YUKSONH,R5

    MOV   BV_TEMP1O,#0 ;kisa devre kontrol icin bu degiskenler
    MOV   BV_TEMP2O,#0 ;sifirlanip kullanılacak
    MOV   BV_TEMP1L,#1 ;deger yollama sayaci olarak kullanılacak
    ;TEMP2H yukselis yonunu belirtir

$BASIC
946 print "***:PH0." YUK%=",YUKO#,YUKL#," DYUK=","DYUK%,"
YUKSON=","YUKSONH#,YUKSONO#,YUKSONL#
948 REM PH0." YUKOFSH=","YUKOFSH#," YUKOFS%=",YUKOFS%
950 REM PH0." YAN%=",YANO#,YANL#," DYAN=","DYAN%,"
YANSON=","YANSONH#,YANSONO#,YANSONL#
952 REM PH0." YANOFSH=","YANOFSH#," YANOFS%=",YANOFS%
$ASM

BASLA:      :*****KONTROL LOOPU BASLANGICI*****
            :*****YAN MOTOR HESABI*****

YANHSP:
    CALL  YANSAYOKU
:HATA%=YANSON%-YAN%
    MOV   R0,BV_YANSONL
    MOV   R2,BV_YANSONO
    MOV   R4,BV_YANSONH
    MOV   R1,BV_YANL
    MOV   R3,BV_YANO
    MOV   R5,#0
    CALL  ISUB2C

```

```

MOV BV_HATAL,R1
MOV BV_HATAO,R3
MOV BV_HATAH,R5

```

```

;EGER HATA BUYUK ISE HATA ISARETINE GORE FULL GAZ VER

```

```

YANYAKLASMASAYI EQU 1;1*256*0.303 milyem=77.58 milyem

```

```

MOV A,BV_HATAH ;HATAH hatanin isaretini HATAH bakarak bul
JB ACC.7,EKSIHT
MOV A,BV_HATAO ;HATA + imis ve HATAO acc al
CLR C
SUBB A,#YANYAKLASMASAYI ;
JC NRMGIT
MOV BV_YYH,#7FH ;+ yonde maksimum hiz
LJMP N991

```

```

EKSIHT: MOV A,BV_HATAO

```

```

CPL A
CLR C
SUBB A,#YANYAKLASMASAYI
JC NRMGIT
MOV BV_YYH,#80H ;- yonde maksimum hiz
LJMP N991

```

```

NRMGIT:

```

```

;HATA YON DEGISTIRINCE HATAYI SIFIRLA

```

```

MOV A,BV_HATAH
XRL A,BV_EYANHATAH
ANL A,#80H ;isaret bitini al
JZ HTCİK
CALL EYANHATA0

```

```

HTCİK:

```

```

;950 YY% = KP# * HATA%

```

```

MOV R0,BV_KP ;R1 R3 R5 daha onceden HATAL,HATAO,HATAH var
MOV R2,#0
MOV R4,#0
CALL IMUL3
MOV R0,#255
CALL IMUL3
MOV BV_YYL,R1
MOV BV_YYO,R3
MOV BV_YYH,R5

```

```

; integral KI

```

```

;HATA%=HATA%+EYANHATA%

```

```

MOV R0,BV_HATAL
MOV R2,BV_HATAO
MOV R4,BV_HATAH
MOV R1,BV_EYANHATAL
MOV R3,BV_EYANHATAO
MOV R5,BV_EYANHATAH
CALL IADD3
MOV BV_HATAL,R1
MOV BV_HATAO,R3
MOV BV_HATAH,R5

```

```
;YY% = YY% + KI% * HATA%
```

```
MOV R0,BV_KI
MOV R2,#0
MOV R4,#0
CALL IMUL3
MOV R0,BV_YYL
MOV R2,BV_YYO
MOV R4,BV_YYH
CALL IADD3
MOV BV_YYL,R1
MOV BV_YYO,R3
MOV BV_YYH,R5
```

```
;EYANHATA# = HATA#
```

```
MOV BV_EYANHATAL,BV_HATAL
MOV BV_EYANHATAO,BV_HATAO
MOV BV_EYANHATAH,BV_HATAH
```

```
;YANPWM HESABI
```

```
N991:
```

```
MOV BV_M,#0
MOV A,BV_YYH
JB ACC.7,EKSI
RL A
MOV BV_YANPWM,A
ORL BV_M,#10H
SJMP CIK995
```

```
EKSI:
```

```
CPL A
RL A
MOV BV_YANPWM,A
ORL BV_M,#20H
```

```
CIK995:
```

```
CALL YUKHSP ;*****
```

```
;$BASIC
```

```
;92 REM PH0." YAN%=",YANO#,YANL#," hata=",HATAH#,HATAO#,HATAL#,"
PWM=",YANPWM#
```

```
;94 PH0." YUK%=",YUKO#,YUKL#," YY=",YYH#,YYO#,YYI#," YUKPWM=",YUKPWM#,"
YON=",YON#," M=",M#
```

```
HIZFLAG BIT 20H.1
```

```
JB HIZFLAG,HIZLIGIT
MOV A,#0E0H
CJNE A,BV_YUKPWM,SOR
```

```
SOR: JC HIZLIGIT ;A<YUKPWM
```

```
;yavas yaklasma
```

```
;PORT1=M#
```

```
MOV P1,BV_M
SETB P1.3 ;busi bosalt
```

```
;puls loopu
```

```
MOV R6,#255
MOV R0,#0
CLR A
```

```

DISLOOP:                                ;loop basi ****
      CJNE  A,BV_YANPWM,GIT1
      ANL  P1,#11001111B                ;YAN puls sondur
GIT1:  CJNE  A,BV_YUKPWM,GIT2
      ANL  P1,#00111111B                ;YUKS puls sondur
GIT2:  INC   A
      MOV  R7,#5                          ;bekleme
      DJNZ R7,$
      CJNE R6,#TUR/2,GIT3
      CALL KSBAK
GIT3:  DJNZ R6,DISLOOP                    ;loop sonu ***
      CLR  P1.3
      JMP  DVM

:HIZLI HAREKET*****
HIZLIGIT:
      MOV  P1,BV_M
      SETB P1.3                          ;busi bosalt

      MOV  A,BV_YANPWM
      JNZ  GITH1
      ANL  P1,#11001111B                ;YAN puls sondur

GITH1: MOV  A,BV_YUKPWM
      JNZ  GITH2
      ANL  P1,#00111111B                ;YUKS puls sondur

GITH2: CALL KSBAK
      CLR  P1.3

DVM:
      DEC  BV_TURSAY                      ;disariya pozisyonu yolla
      MOV  A,BV_TURSAY
      JNZ  DEVAM2
      MOV  BV_TURSAY,#TUR/2              ;her TUR pulsta bir buraya gelir

      CALL PAYAR

      ;CALL YUKHIZOKU

      MOV  R1,#'D'                        ;ve D harfi yollar sonra deger yollar
      MOV  R2,BV_YANO
      MOV  R3,BV_YANL
      MOV  R4,BV_YUKO
      MOV  R5,BV_YUKL
      MOV  R6,BV_GP
      MOV  R7,BV_SAR
      MOV  R0,BV_TEMP1L;sayac taki sayi hangi degerin yollanacagini
      MOV  A,@R0                          ;soyler
      INC  BV_TEMP1L
DVM1:  CLR  TI
      MOV  SBUF,A
      CJNE R0,#7,DEVAM2 ;7 adet deger yollandimi

```

```

MOV BV_TEMP1L,#1 ; deger yollama sayacini sifirla
DEVAM2:
;herhangi bir giris varmi bak
JNB RI,DEVAM
CLR RI
MOV A,SBUF
CJNE A,#'9',DEVAM
SJMP LN1000 ;Eger 9 geldiyse servo loop tan cik
DEVAM:
LJMP BASLA ;servo loop basina don
$BASIC
1000 GOSUB 9998:RETURN :REM motorlari durdur ventilleri bosalt ve cik
$ASM
;kisadevre bak
KSBAK: MOV R1,A ;ACC degerini sakla
MOV DPTR,#IV_A
MOVX A,@DPTR
MOV P2,A
MOV R0,#0
MOVX A,@R0
ANL A,#00001100B ;XBY(A%) oku ve maskele
JNZ KISADVR ;kisadevre varsa yap
MOV A,BV_TEMP1O ;kisadevre yok
JZ KS3 ;kisa devre yoksa sayilari dusur
DEC BV_TEMP1O
KS3: MOV A,BV_TEMP2O
JZ KSCIK ;O sa c□k
DEC BV_TEMP2O
MOV BV_SAR,#0
SJMP KSCIK

KISADVR: JB ACC.2,YANKSDVR
JB ACC.3,YUKKSDVR
SJMP KSCIK

YANKSDVR:
INC BV_TEMP1O
INC BV_TEMP1O
MOV A,BV_TEMP1O
CJNE A,#250,YANKS1
YANKS1: JC KSCIK ;A<250
;ANL P1,#11001111B ;A>250 yan puls sondu
MOV BV_SAR,#1 ;yan motor ariza
SJMP KSCIK

YUKKSDVR:
INC BV_TEMP2O
INC BV_TEMP2O
MOV A,BV_TEMP2O
CJNE A,#250,YUKKS1 ;
YUKKS1: JC KSCIK ;A<250
;ANL P1,#00111111B ;A>250 yuks puls sondu
MOV BV_SAR,#2 ;yuks motor ariza

```

```
KSCIK:      MOV   A,R1
KSCIK2:     RET
```

```
YUKHSP:     ;***** YUKS. MOTOR HESABI*****
```

```
      CALL  YUKSAYOKU
;HATA%=YUKSON%-YUK%
      MOV   R0,BV_YUKSONL
      MOV   R2,BV_YUKSONO
      MOV   R4,BV_YUKSONH
      MOV   R1,BV_YUKL
      MOV   R3,BV_YUKO
      MOV   R5,#0
      CALL  ISUB2C
      MOV   BV_HATAL,R1
      MOV   BV_HATAO,R3
      MOV   BV_HATAH,R5
;EGER HATA YAVASLAMASAYI DAN BUYUK ISE HATA ISARETINE GORE FULL GAZ VER
YUKYAVASSAYI EQU 25      ;10*0.888888 milyem
      MOV   A,BV_HATAH      ;HATAH hatanin isaretini HATAH bakarak bul
      JB    ACC.7,EKSIHT2
      MOV   A,BV_HATAO      ;HATA + imis ve HATAO acc al
      JNZ  CIK3            ;hata 256 dan büyük
      CJNE R1,#YUKYAVASSAYI,NOT_EQU
      JMP   YAVASGIT        ;R1=YUKYAVASSAYI
NOT_EQU:JC  YAVASGIT        ;R1<YUKYAVASSAYI
CIK3:  MOV  BV_YYH,#7FH      ;R1>YUKYAVASSAYI+ yonde maksimum hiz
      SETB HIZFLAG          ;hizli git
      JMP  YUKPWMHSP
EKSIHT2: MOV A,BV_HATAO      ;HATA - imis ve HATAO acc al
      CPL  A
      JNZ  CIK4            ;hata 256 dan büyük
      MOV  A,BV_HATAL
      CPL  A
      CJNE A,#YUKYAVASSAYI,NOT_EQU2
      JMP  YAVASGIT        ;A=YUKYAVASSAYI
NOT_EQU2:JC YAVASGIT        ;A<YUKYAVASSAYI
CIK4:  MOV  BV_YYH,#80H      ;A>YUKYAVASSAYI- yonde maksimum hiz
      SETB HIZFLAG          ;hizli git
      JMP  YUKPWMHSP

YAVASGIT:
      CLR  HIZFLAG
;HATA YON DEGISTIRINCE HATAYI SIFIRLA
      MOV  A,BV_HATAH
      XRL  A,BV_EYUKHATAH
      JNB  ACC.7,HTCIK2
      CALL EYUKHATA0
HTCIK2:
;950 YY% = KP# * HATA%
      MOV  R0,BV_KP      ;R1 R3 R5 daha onceden HATAL,HATAO,HATAH var
      MOV  R2,#0
      MOV  R4,#0
```

```

CALL IMUL3
MOV R0,#255
CALL IMUL3
MOV BV_YYL,R1
MOV BV_YYO,R3
MOV BV_YYH,R5
; integral KI
:HATA%=HATA%+EYUKHATA%
MOV R0,BV_HATAL
MOV R2,BV_HATAO
MOV R4,BV_HATAH
MOV R1,BV_EYUKHATAL
MOV R3,BV_EYUKHATAO
MOV R5,BV_EYUKHATAH
CALL IADD3
MOV BV_HATAL,R1
MOV BV_HATAO,R3
MOV BV_HATAH,R5
:YY% = YY% + KI% * HATA%
MOV R0,BV_KI
MOV R2,#0
MOV R4,#0
CALL IMUL3
MOV R0,BV_YYL
MOV R2,BV_YYO
MOV R4,BV_YYH
CALL IADD3
MOV BV_YYL,R1
MOV BV_YYO,R3
MOV BV_YYH,R5

:EYUKHATA% = HATA%
MOV BV_EYUKHATAL,BV_HATAL
MOV BV_EYUKHATAO,BV_HATAO
MOV BV_EYUKHATAH,BV_HATAH

YUKPWMHSP:
MOV A,BV_YYH
JB ACC.7,EKSI2
RL A ;yyh +
MOV BV_YUKPWM,A
ORL BV_M,#40H
MOV BV_YON,#1 ;yon 1
RET
EKSI2: CPL A ;yyh -
RL A
MOV BV_YUKPWM,A
ORL BV_M,#80H
MOV BV_YON,#0 ;yon 0
RET

:HESAP SUBROUTINLERI

```

YUKSAYOKU:

```

    ORL    P1,#00000001B
    ANL    P1,#11110001B
; YUKO#=XBY(B%):YUKL#=XBY(A%)
    MOV    DPTR,#IV_B
    MOVX   A,@DPTR
    MOV    R0,#0
    MOV    P2,A
    MOVX   A,@R0
    MOV    BV_YUKO,A

```

```

    MOV    DPTR,#IV_A
    MOVX   A,@DPTR
    MOV    P2,A
    MOVX   A,@R0
    MOV    BV_YUKL,A

```

;varsa-offseti ekle

```

    MOV    DPTR,#IV_YUKOFS
    CALL   IGETVAR
    MOV    R1,BV_YUKL
    MOV    R3,BV_YUKO
    CALL   IADD
    MOV    BV_YUKL,R1
    MOV    BV_YUKO,R3
    RET

```

;R0 ,R2 da YUKOFS var

YANSAYOKU:

```

    ANL    P1,#11110000B ;yan sayiciyi okumak icin
; YANO#=XBY(B%):YANL#=XBY(A%)
    MOV    DPTR,#IV_B
    MOVX   A,@DPTR
    MOV    R0,#0
    MOV    P2,A
    MOVX   A,@R0
    MOV    BV_YANO,A

```

```

    MOV    DPTR,#IV_A
    MOVX   A,@DPTR
    MOV    P2,A
    MOVX   A,@R0
    MOV    BV_YANL,A

```

;offset varsa ekle

```

    MOV    DPTR,#IV_YANOFS
    CALL   IGETVAR
    MOV    R1,BV_YANL
    MOV    R3,BV_YANO
    CALL   IADD
    MOV    BV_YANL,R1
    MOV    BV_YANO,R3
    RET

```

;R0,R2 da YANOFS var

IADD3:

;R0,R2,R4 sayisini R1,R3,R5 sayisi ile toplar

```

;netice R1,R3,R5 kalir
CLR PS.2
CLR C
MOV A,R1
ADDC A,R0
MOV R1,A

MOV A,R3
ADDC A,R2
MOV R3,A

MOV A,R5
ADDC A,R4
JB PSW.2,OVFLOW ;2 positif sayi toplama neticesi negatif
MOV R5,A ;2 negatif sayi toplami neticesi pozitif
RET
OVFLOW: MOV A,R5
JB ACC.7,EKSMX
JMP ARTMX

```

:R5,R3,R1 ILE R0 CARPAR NETICESINI R5,R3,R1 E YAZAR

:overflow varsa maksimum - veya + sayiyi koyar

IMUL3:

```

MOV A,R5
MOV R6,A ;MSB yi R6 ya al
MOV B,R1
MOV A,R0
MUL AB
MOV R1,A
MOV TMP_1,B

MOV B,R3
MOV A,R0
MUL AB
CLR C
ADD A,TMP_1
MOV R3,A
JNC M1
INC B
M1: MOV TMP_1,B

MOV B,R5
MOV A,R0
MUL AB
CLR C
ADD A,TMP_1
JNC M2
INC B
M2: MOV R5,A
;overflow kontrol
MOV A,R6 ;ilk gelen sayi
JB ACC.7,EKSH1 ;ilk gelen sayi eksi ise

```

```

ARTI:  MOV  A,B           ;ilk gelen sayi arti imis
       JNZ  ARTMX       ;sayi tasmis artimax set et
       MOV  A,R5
       JNB  ACC.7,OKCIK ;carpim neticesi +ise OK ve cik
ARTMX:
       MOV  R5,#07FH    ;- ise set +MAX
       MOV  R3,#0FFH
       MOV  R1,#0FFH
       RET

EKSI1:           ;ilk gelen sayi eksi imis
       MOV  A,R5
       JB   ACC.7,OKCIK ;carpim neticesi- ise OK ve cik
EKSMX:           ;sayi tasmis eksimax set et
       MOV  R5,#080H    ;+ ise set -MAX
       MOV  R3,#00H
       MOV  R1,#00H
OKCIK: RET

ISUB2C:           ;cikartma icin ikinci sayinin
       CLR  C           ;2'complement al ve add
       MOV  A,R1
       CPL  A           ;1's complement
       ADDC A,#1       ;2's complement
       MOV  R1,A

       MOV  A,R3
       CPL  A
       ADDC A,#0
       MOV  R3,A

       MOV  A,R5
       CPL  A
       ADDC A,#0
       MOV  R5,A

       CALL IADD3
       RET

EYANHATA0:       ;eski yan hatayi sifirlar
       MOV  BV_EYANHATAL,#0
       MOV  BV_EYANHATAO,#0
       MOV  BV_EYANHATAH,#0
       MOV  BV_TEMP1O,#0 ;kisa devre kontrol icin bu degiskenler
       RET

EYUKHATA0:       ;eski yuks hatayi sifirlar
       MOV  BV_EYUKHATAL,#0
       MOV  BV_EYUKHATAO,#0
       MOV  BV_EYUKHATAH,#0
       MOV  BV_TEMP2O,#0 ;kullanilacak
       RET

DGRYOLLA:

```

```
MOV  A,BV_YANO
MOV  R6,A           ;chksum icin
CALL YOLLA

MOV  A,BV_YANL
ADD  A,R6
MOV  R6,A
MOV  A,BV_YANL
CALL YOLLA

MOV  A,BV_YUKO
ADD  A,R6
MOV  R6,A
MOV  A,BV_YUKO
CALL YOLLA

MOV  A,BV_YUKL
ADD  A,R6
MOV  R6,A
MOV  A,BV_YUKL
CALL YOLLA

MOV  DPTR,#IV_EGX
CALL IGETVAR       ;R0 da EGXL R2 de EGXO

MOV  A,R2
ADD  A,R6
MOV  R6,A
MOV  A,R2
CALL YOLLA

MOV  A,R0
ADD  A,R6
MOV  R6,A
MOV  A,R0
CALL YOLLA

MOV  DPTR,#IV_EGY
CALL IGETVAR       ;R0 da EGYL R2 de EGYO

MOV  A,R2
ADD  A,R6
MOV  R6,A
MOV  A,R2
CALL YOLLA

MOV  A,R0
ADD  A,R6
MOV  R6,A
MOV  A,R0
CALL YOLLA

MOV  A,BV_GP
```

```

        ADD    A,R6
        MOV    R6,A
        MOV    A,BV_GP
        CALL   YOLLA

        MOV    A,BV_SAR
        ADD    A,R6
        MOV    R6,A
        MOV    A,BV_SAR
        CALL   YOLLA

        MOV    A,#0FFH           ;chksum hesap et
        CLR    C                 ;chksum yolla
        SUBB   A,R6
        INC    A
        CALL   YOLLA

        RET

YOLLA:   JNB    TI,$
        CLR    TI
        MOV    SBUF,A
        CALL   BEKLE
        RET

BEKLE:   MOV    R7,#255          ;24*90.4 ns=2.1 us
        DJNZ   R7,$            ;24*90.4 ns=2.1 us
        RET                    ;24*90.4 ns=2.1 us 4.2*256+2.1=1.075 ms

BEKLE4:
        CALL   BEKLE
        CALL   BEKLE
        CALL   BEKLE
        CALL   BEKLE
        RET

SARYOLLA:
        MOV    A,#'D'           ;ve D harfi yollar sonra deger yollar
        CALL   YOLLA

        MOV    A,BV_YANO
        CALL   YOLLA

        MOV    A,BV_YANL
        CALL   YOLLA

        MOV    A,BV_YUKO
        CALL   YOLLA

        MOV    A,BV_YUKL
        CALL   YOLLA

```

```

                MOV  A,BV_GP
                CALL YOLLA

                MOV  A,BV_SAR
                CALL YOLLA

DNI:            MOV  R6,#50
                CALL BEKLE
                DJNZ R6,DNI
                RET

SIFIRLA:
                MOV  BV_SAR,#0
                MOV  P1,#08H
                MOV  DPTR,#IV_C
                CALL IGETVAR
                MOV  DPL,R0
                MOV  DPH,R2
                MOV  A,#0
                MOVX @DPTR,A

                MOV  P1,#4
                CALL BEKLE
                MOV  P1,#0001000B
                MOV  BV_M,#0
                RET

KONSOLDANDGROKU:
;INPUT DYAN%,DYUK%
                MOV  R7,#0                ;chksum icin

                JNB  RI,$
                MOV  A,SBUF
                CLR  RI
                MOV  BV_TEMP10,A
                ADD  A,R7
                MOV  R7,A

                JNB  RI,$
                MOV  A,SBUF
                CLR  RI
                MOV  BV_TEMP11,A
                ADD  A,R7
                MOV  R7,A

                JNB  RI,$
                MOV  A,SBUF
                CLR  RI
                MOV  BV_TEMP20,A
                ADD  A,R7
                MOV  R7,A

                JNB  RI,$

```

```

MOV  A,SBUF
CLR  RI
MOV  BV_TEMP2L,A
ADD  A,R7
MOV  R7,A

JNB  RI,$           ;checksum mesaji
MOV  A,SBUF
CLR  RI
ADD  A,R7
JZ   DCIK          ;checksum kontrolu yap yanlis ise SAR=6
MOV  BV_SAR,#6     ;iletisim hatasi
DCIK:
RET

HFZTST:
MOV  DPTR,#0
RAMTST:
;MOVX A,@DPTR
;MOV  R4,A
;MOV  A,#5AH
;MOVX @DPTR,A
;MOVX A,@DPTR
;CJNE A,#5AH,RMTST2 ;Dogru degilse ariza ver ve cik
;MOV  A,R4
;MOVX @DPTR,A
;MOV  A,DPH
;CJNE A,#0FH,RAMTST ;8K ram =01FH 'e kadar tara
ROMTST:
MOV  DPTR,#0
MOV  R1,#0
RMTST1: CLR  A
MOV  A,@A+DPTR
ADD  A,R1
MOV  R1,A
INC  DPTR
MOV  A,DPH
CJNE A,#040H,RMTST1 ;ROM 16K ise 040H
;CJNE R1,#00,RMTST2 ;chksum degerine bak
RET
RMTST2: MOV  BV_SAR,#7
RET

ELLEPAYAR:
;basinc istenen duzeyde ise ACC=0 yapar
DEC  BV_TURSAY
MOV  A,BV_TURSAY
JNZ  BSNCYANLIS
MOV  BV_TURSAY,#TUR/2

CALL YUKSAYOKU
CALL ADCBASLAT ;ADC baslat

```

```

CALL BEKLE4
CALL PAYAR

MOV A,BV_V ;bak basinc dogrumu
ANL A,#00000011B
JZ BSNCOK ;0 sifirsa ventiller kapali
;basinc dogru demektir
BSNCYANLIS: MOV A,#255 ;basinc ayarlanamamis ise
RET

BSNCOK: MOV A,#0 ;basinc tamam ACC=0
RET

ADCBASLAT:
;XBY(C%)=0
MOV DPTR,#IV_C
MOVX A,@DPTR
MOV R2,A
MOV R0,#0
MOV A,#0 ;c portuna basinc icin 0 adr yaz
MOV P2,R2
MOVX @R0,A
ORL P1,#00000111B ;REM ADC start pulsu
ANL P1,#11110111B
NOP
NOP
ORL P1,#00001000B
ANL P1,#11111000B
RET

ADCBSNCOKU:
;PORT1=6 ADC oe
ORL P1,#00000110B ;bir onceki ADC degerini oku
ANL P1,#11110110B
;GP#=XBY(A%)
MOV DPTR,#IV_A
MOVX A,@DPTR
MOV R0,#0
MOV P2,A
MOVX A,@R0 ;ACC de okunan basinc var
;CLR C
;ADDC A,BV_GP ;basinc suzgeci
;RRC A ;2 ye bol
MOV BV_GP,A ;basinci yaz
RET

:****BASINC AYAR

PAYAR:
CALL ADCBSNCOKU

MOV A,BV_YUKL
ANL A,#11110000B ;ust digitleri al
SWAP A

```

```

MOV R7,A ;bv-YUKL nin ust nibble

MOV A,BV_YUKO
ANL A,#00001111B
SWAP A
ADD A,R7

:yukselis yonune gore degisiklik yap
MOV R7,BV_YON ;hareket yonu 1 se yukselir
CJNE R7,#0,CIKARKEN

:inerken
MOV DPH,BV_INBSNCADR ;alcalis
SJMP DVM8 ;look up table adresi
CIKARKEN: ;yukaricikacak
MOV DPH,BV_CIKBSNCADR ;yukselis look up table adresi
DVM8:
MOV DPL,A
MOVX A,@DPTR ;look up tabledeki basinc dgr oku
:HATA hesabi
CLR C
SUBB A,BV_GP
:hatanin yonune gore art ventili ac veya dus ventili ac
:veya 0 sa ventilleri durdur
JZ KAPATCIK ;basinc istenen basinca esitse
JC DUS ;cikartma neticesi - ise dus ventili ac

ART: ;cikartma neticesi + ise art ventili ac
ORL BV_V,#00000001B
SJMP PLSVER

DUS: ;cikartma neticesi - dus ventilini ac
ORL BV_V,#00000010B
SJMP PLSVER

KAPATCIK: ANL BV_V,#0 ;ventilleri kapat
; ventile puls ver XBY(C%)=V#
PLSVER: MOV DPTR,#IV_C
MOVX A,@DPTR
MOV R2,A
MOV R0,#0
MOV A,BV_V
ANL A,#00000011B
CJNE A,#3,ATLA ;iki ventil beraber cekmemeli
ANL BV_V,#11111100B
MOV A,#0
ATLA: MOV A,BV_V
MOV P2,R2
MOVX @R0,A

:PORT1 04H ;ventil set edilsin diye
ORL P1,#00000100B
ANL P1,#11110100B
NOP
NOP

```

```

        CLR    P1.2
;PC portunu 0 da birak adc icin
        MOV    A,#0
        MOV    P2,R2
        MOVX   @R0,A

```

```

        ORL    P1,#00000111B      ;REM ADC start pulsu
        ANL    P1,#11110111B
        NOP
        NOP
        ORL    P1,#00001000B
        ANL    P1,#11111000B
        RET

```

YANSIFIRLA:

```

        MOV    P1,#0
        MOV    P1,#2
        CALL   BEKLE
        MOV    P1,#0
        RET

```

YUKSIFIRLA:

```

        MOV    P1,#0
        MOV    P1,#3
        CALL   BEKLE
        MOV    P1,#0
        RET

```

\$BASIC

```

        REM ***** tum sistemi sifirla

```

9998 REM

\$ASM

```

        CALL   SIFIRLA

```

\$BASIC

9999 RETURN

11000 END