

**Fırat Üniversitesi**  
**Mühendislik Fakültesi**  
**Elektrik-Elektronik Mühendisliği Bölümü**  
**Bitirme Projesi**

**Hazırlayan**  
**Veli Özerdem**

**Öğretim Üyesi**  
**Hasan H. Balık**

# İçindekiler

## Giriş

## Konu

## SCADA nedir?

## Bir SCADA sistem uygulaması

### A. Algılayıcılar

#### A.1. Ultrasonik algılayıcı devreler

#### A.2. Sıcaklık sensörü

### B. Sürücü devresi

### C. Ara birim

#### C.1. Haberleşme standardı seçimi

#### C.2. Asenkron seri haberleşme standardı

#### C.3. Pic16F877

#### C.4. Pic16F877 Usart

#### C.5. Pic16F877 Analog dijital çevrici birimi

#### C.6. Arabirim program akış şeması

#### C.7. Pic-PC uyarlayıcı

### D. PC ortamı ve Arayüz programı

#### D.1. Ana program akış şeması

## Giriş

Üretim teknolojileri büyüyüp geliştikçe sistemlerin kontrolü ve gözlenebilirliği daha da önem kazanmaktadır. Üretim sürecindeki her bir sistemin anında kontrol edilmesi ve istenilen performans kriterlerine uygun üretim yapılması ve bunun eş zamanlı gerçekleşmesi problemleri, elektrik, elektronik, elektromekanik, hidrolik vb sistemler ile bilgisayar teknolojisinin uygun arabirimler vasıtası ile bir araya getirilmesiyle çözülmektedir. Öyle ki; büyük ölçekli bir üretim faaliyeti bile bir bilgisayarın çevresel bir aygıtı gibi kontrol edilebilmektedir. Elbette böyle bir tümleşik yapı, birden fazla mühendislik dalının ortak çalışmasını gerektirir. (bilgisayar, haberleşme, kontrol mühendisliği gibi.)

Sanayi devriminde gelişen makineleşme ürün standartlarını ve miktarını artırmıştır. Ancak kompleks üretim birimlerindeki verim artışı halen devam etmektedir. Bu noktada karşımıza doğru verinin doğru yerde bulunması ihtiyacı çıkmaktadır. Büyük bir üretim bandındaki olası durumların, hataların ve tüm sürecin tek bir arabirim sayesinde tek bir noktada görülebilmesinin, daha doğru karar verebilme, sonuca daha kısa zamanda ulaşabilme gibi getirileri olacaktır. Bu noktada devreye SCADA sistemler girmekte ve iş yükünün büyük bir kısmını hantal iş gücünden alıp elektronik omuzlara bırakmaktadır. *Supervisory Control and Data Acquisition* anlamına gelen scada üretim sistemlerindeki verimlilik artışı ve modernizasyonun bu gün itibarı ile geldiği son noktadır. Ben de bitirme ödevimi, içerisinde mesleğimle ilgili bir çok konuyu barındıran bu sistemler üzerine hazırlayarak, eğitmenlerimin dimağında inşa ettikleri temele sağlam bir tuğla daha koymayı amaçlıyorum. Proje sürecinde birikimlerinden ilham aldığım öncelikle proje sorumlusu Doç. Dr. Hasan BALIK'a, ve Arş. Gör. Yavuz Erol'a teşekkür ederim.

**Konu.....:** Bir otomasyon sisteminin kontrolü ve bilgisayarda eş zamanlı simule edilmesi

## SCADA nedir?

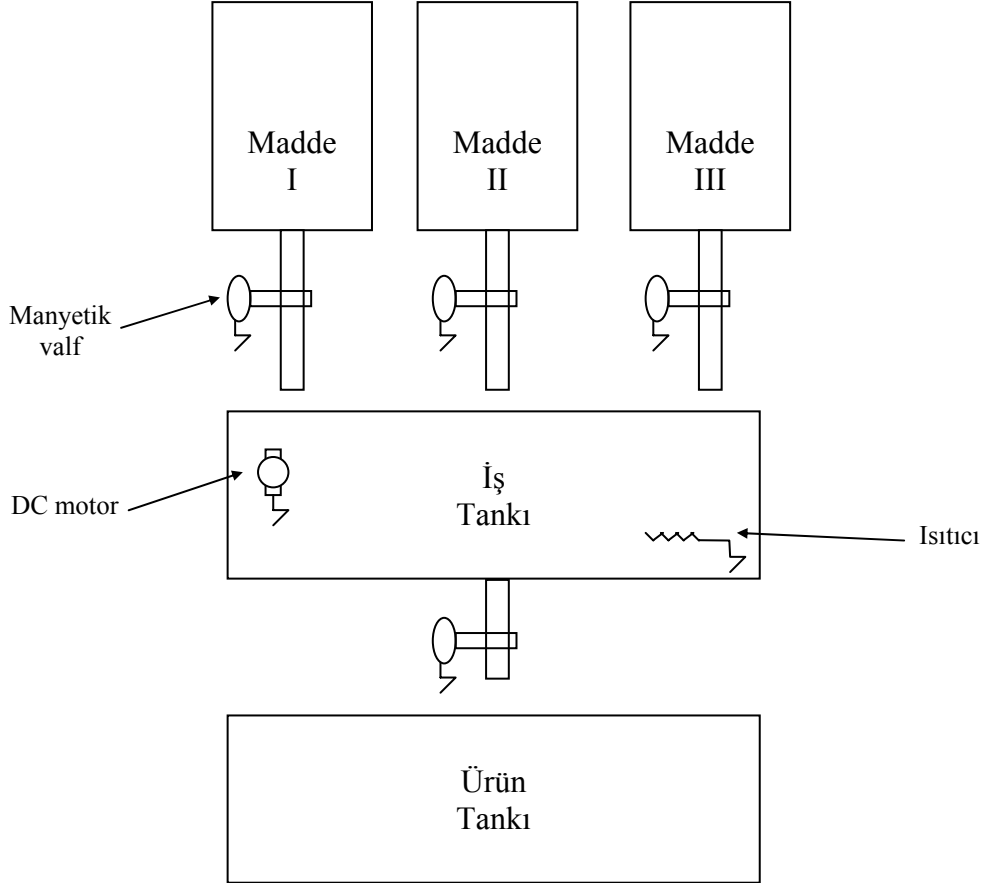
SCADA omurgalı bir sistem, işlemi, algılayıcılar (akım, gerilim, adet, sıcaklık, seviye, basınç, hız, tork vb) vasıtası ile görür. Bu değerler belli formatlarda bilgisayara aktarılır. İşin buraya kadarki kısmı donanımla ilgilidir. Bundan sonra yazılımın devreye girmesiyle kullanıcıya bu değerlerden elde edilen görsel bir ortam sunulur. Bu data akışı çift yönlüdür. Öyle ki; örneğin kullanıcının bir mouse hareketi ile rahatlıkla devasa büyüklükteki bir türbinin kapakçıklarına yön verilebilir. Bu işlemden sonra akan suyun debisi, basıncı yine aynı ekrandan izlenebilir. Sistem belli hata kodlarını sürekli tarayarak acil durum uygulamaları yapabilir. Özellikle insan hayatı açısından olumsuz şartlara sahip ortamlarda (soda, gübre fabrikaları gibi) güvenliği sağlar ve bu sayede tek noktadan kontrolün avantajlarını bize sunar.

SCADA sistemi operatörler için ileri düzeyde kontrol ve gözetleme özellikleri sağlamaktadır. Genel olarak SCADA sistemi, uygulamada şu imkanları sağlayabilir.

- Kullanıcı tarafından tanımlanmış işletmeye ait parametreler (seviye, sıcaklık, basınç, dijital sinyaller, vana ve motor durumları, sistem durumu vb.) vasıtasıyla işletmenin takibi
- Reçete ekranları vasıtasıyla, üretim reçetelerinin girilmesi ve işleyen reçeteler hakkında operatörün bilgilendirilmesi
- Parametre ekranları vasıtasıyla, sistem için gerekli olan limit değerlerin (set-point, alt ve üst alarm değerleri) girilmesi
- P, I, D parametrelerinin girilebilmesi ve gözetlenmesi
- İşletme değerlerinin tarihsel ve gerçek zamanlı trendlerinin tutulması
- Anlık ve periyodik raporların (üretim, reçete, stok vb.) alınması
- Otomatik çalışan sisteme, SCADA ekranlarından manuel müdahale imkanı
- Alarm ve durumların gösterilmesi ve yazıcıya ve/veya veri tabanına kayıt edilmesi
- İleri düzeyde kalite kontrol (örneğin istatistiksel proses kontrol) desteği

## Bir Scada Sistem Uygulaması

Bu uygulamada 3 farklı maddenin istenilen oranlarda, istenilen sıcaklık ve süre karıştırılarak yeni bir ürün elde edilmesini ve bu sürecin bilgisayar ekranından kontrolünü ele alacağız. Sistem şeması *şekil 1*'deki gibidir. Arabirim, bu süreç içerisinde PC ile üretim sistemi arasındaki iletişimi gerçekleştirecektir. PC-Arabirim arasındaki iletişim, bilgisayar haberleşme standartlarından biri olacağı için daha sonra değineceğim.



Şekil 1: Üretim sistemi şematik gösterimi

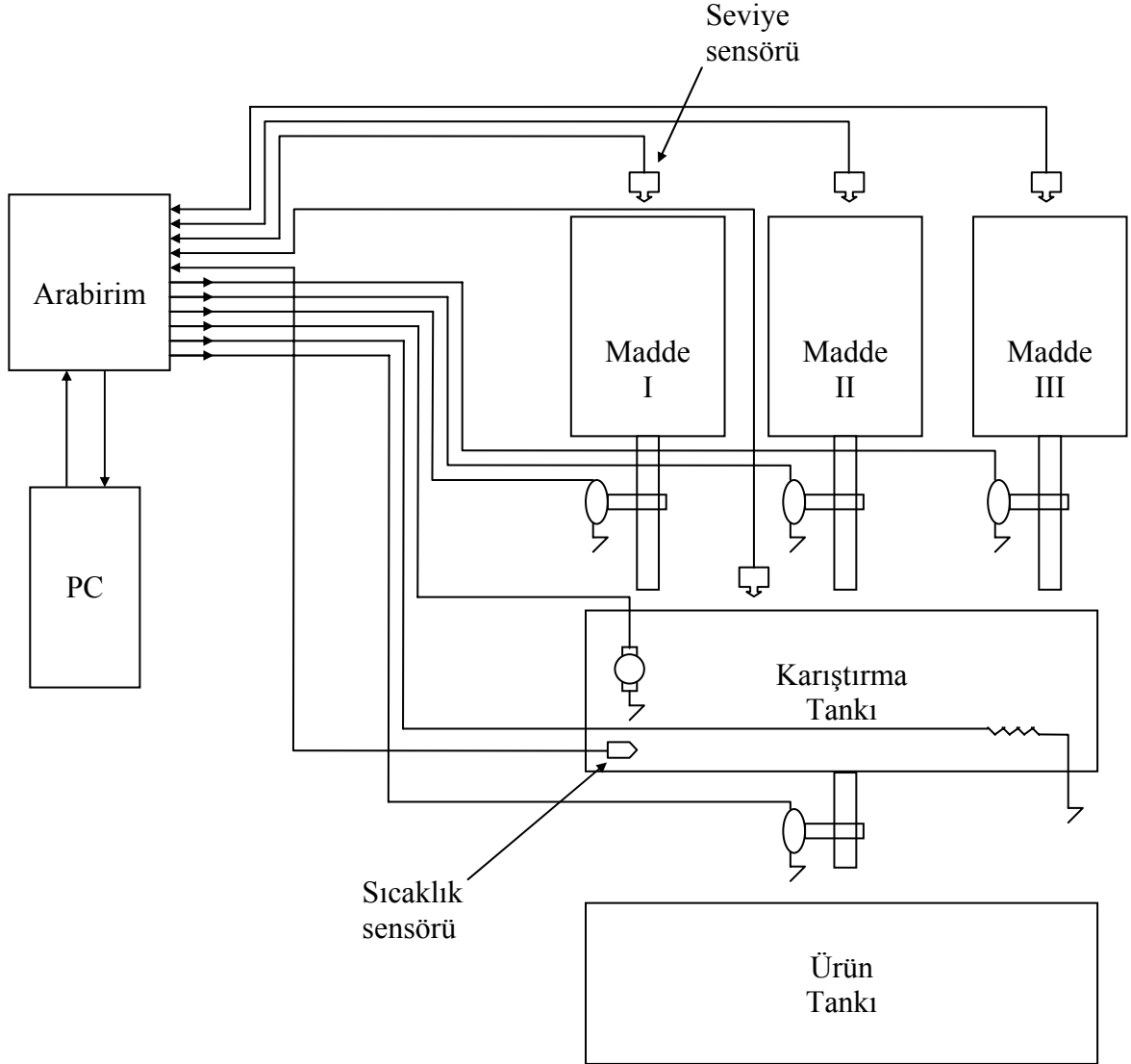
Arabirim-Sistem arasındaki veri ve kontrol sinyalleri akışı için önce nelerin kontrol edileceğini ortaya koyalım.

Hammaddelerin ve ürünün sevki ve seviyesi, ürün hazırlama tankının sıcaklığı ve karıştırma motoru kontrol edilecektir. Dolayısıyla sistemdeki elektrik enerji domeninde yer alan elemanları, bağlı oldukları fiziksel objeye göre tablo halinde şu şekilde gösterebiliriz:

Eleman	1. Tnk	2. Tnk	3. Tnk	İş Tankı	Akış Yönü	Data Tipi
Seviye sensörü	√	√	√	√	Giriş	Boolean
Manyetik valf	√	√	√	√	Çıkış	Boolean
Isıtıcı	-	-	-	√	Çıkış	Boolean
DC motor	-	-	-	√	Çıkış	Boolean
Sıcaklık Sensörü	-	-	-	√	Giriş	Analog

Tablo 1: Eleman konum veri akış-tip tablosu

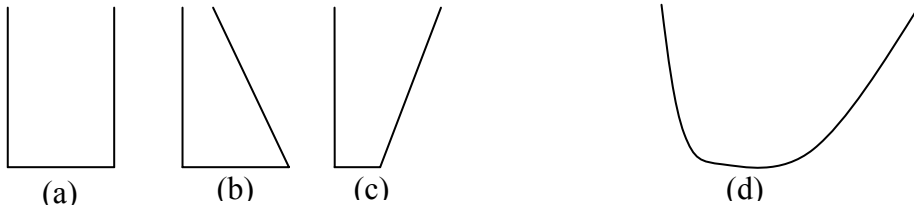
Tabloda *çıkış* olarak belirtilmiş elemanlar zaten uygulamayı düşündüğümüz sistemde var olan yada bize kontrol etmemiz için verilen sistemde bulunanlardır. Bunları kontrol etmek için hem açık çevrim hemde kapalı çevrim kontrol sistemi tercih edilmesi söz konusudur. Ancak açık çevrim kontrol sistemi için hassas zamanlama, debi vb. kalibrasyonları gerekir. Ve bu sadece bu sistem için geçerli olur. Fiziksel sistem değiştiğinde bütün bu değerlerin yeni sisteme göre tekrar kalibre edilmeleri gerekir. Ancak kapalı çevrim kontrolü seçersek, fiziksel sistem boyut olarak değişse bile, sadece ayar noktaları değiştirilerek yeni sisteme adapte edilebilir. Tabi bunun için de geri besleme elemanlarına ihtiyacımız olacaktır. Tabloda *giriş* olarak belirtilen elemanlar bu ihtiyacı karşılamak için seçilmiştir. Sistemde karıştırma işi dışındaki tüm çıkışlar için (valf ve ısıtıcı) bir geri besleme elemanı (seviye ve sıcaklık sensörü) kullanılmıştır. Ve bu elemanların kontrolü geri beslemeli aç-kapa (on-off feedback) ile yapılmaktadır. Kontrol işlemi yazılımla ve anlık yapılacağından, sıcaklık kontrolü dışındaki kontrol işlemlerine belli bir histeresiz eklenmesine gerek duyulmaz. Sistem donanımı tüm hali ile *şekil 2*'de gösterilmiştir.



Şekil 2: Scada sistemi donanım şematik gösterimi

## A. Algılayıcılar

Kullanılacak madde miktarını elektronik olarak kontrol edebilmek için çeşitli yöntemler düşünülebilir. Ancak endüstride en çok kullanılan yöntem, özellikle maddelerin içinde buldukları kap (silo, depo vs.) yükseltiye bağlı bir tek formülle ifade edilebilen hacim değişimi gösteriyorsa seviye ölçme yöntemidir. Sonuçta madde miktarı hacimleri ölçülerek hesaplanır. Şekil 3'te (a), (b), (c) ile gösterilenler bu gruba girer. (d)'nin ise madde miktarını ölçmek için seviye bilgisi kullanmak fonksiyonel olmayacaktır. Çünkü seviyeye bağlı matematiksel bir cümle ile hacim hesabı yapmak mümkün olmamaktadır. Ancak yine de (d) türü silo için seviye tespit yöntemi kullanılacaksa, her yükseltiye ait madde miktarını içeren bir değer tablosu kullanarak formül kullanmadan yapılabilir. Seviye verileri ne kadar yoğunsa (yüksek çözünürlük) tablo verileri de aynı oranda yoğun olmalıdır.



Şekil 3: silo şekil tipleri

Bu sistemde seviye ölçerek madde miktarı tespit yöntemi kullanıldı. Seviye ölçmek içinse yine çeşitli yöntemler mevcut. Bunlar ölçülecek maddenin cinsine göre değişiyor. Örneğin çimento üretim tesislerinde ultrasonik yada mekanik yöntemler kullanılıyor. Maddenin iletkenliğinden faydalanılarak yapılan seviye tespit yöntemleri de bir çok dalda kullanılıyor. Bu sistemde ultrasonik ses ile seviye tespit yöntemi kullanıldı. Sıcaklık göreceli bir büyüklük olmadığından ölçme tekniği ve sensör seçimi üzerinde durmayacağım.

### A.1. Ultrasonik Seviye Dedektörü

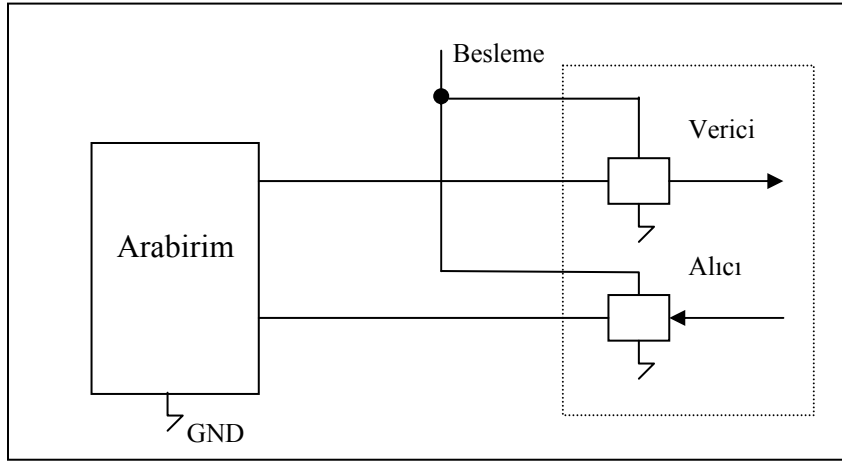
Ultrasonik kelime anlamı olarak insan kulağının duyma sınırının hemen üzerindeki ses dalgalarına verilen isimdir. Bu dalgaların frekans aralığı 20-500 kHz dir. Ultrasonik algılayıcıların temel çalışma mantığı, ultrasonik ses dalgasının, ultrasonik verici tarafından üretilerek gönderilmesi ve uzaktaki bir nesneden yansıyan sesin alıcı tarafından alınarak tekrar elektriksel işarete çevrilmesi esasına dayanır. Gönderme ve alma arasında geçen sürede ses dalgası, cisim ile aradaki mesafeyi 2 kez kat eder. Ultrasonik sesin havada yayılma hızı 340 m/s'dir. Dolayısıyla aradaki mesafeyi basit bir işlemle hesaplamak mümkündür. Sesin gidiş-geliş için harcadığı süre  $\Delta t$  ve aradaki mesafe  $\Delta x$  ise:

$$\Delta X = \frac{\Delta t}{2} \cdot V_{ses} = \frac{\Delta t}{2} \cdot 340 \text{ m/sn}$$

olacaktır. Şekil 4'te ilgili şematik. Şekil 5'te ise ultrasonik algılayıcı ile arabirim bağlantılarını gösterilmiştir.



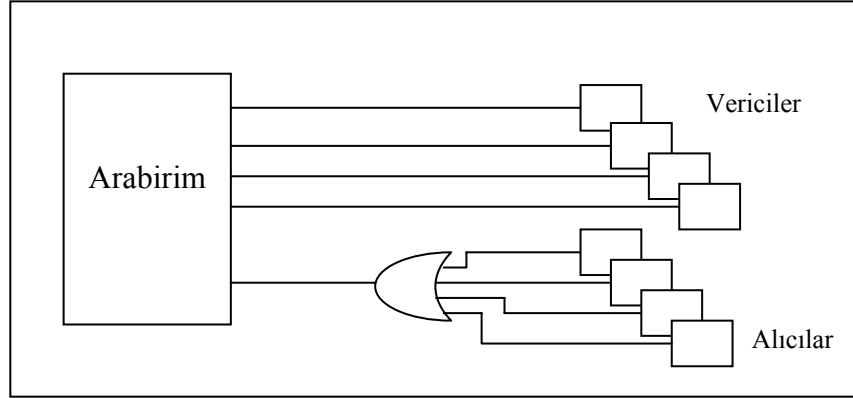
Şekil 4: Ultrasonik mesafe algılama



Şekil 5: Ultrasonik algılayıcı bağlantı şeması

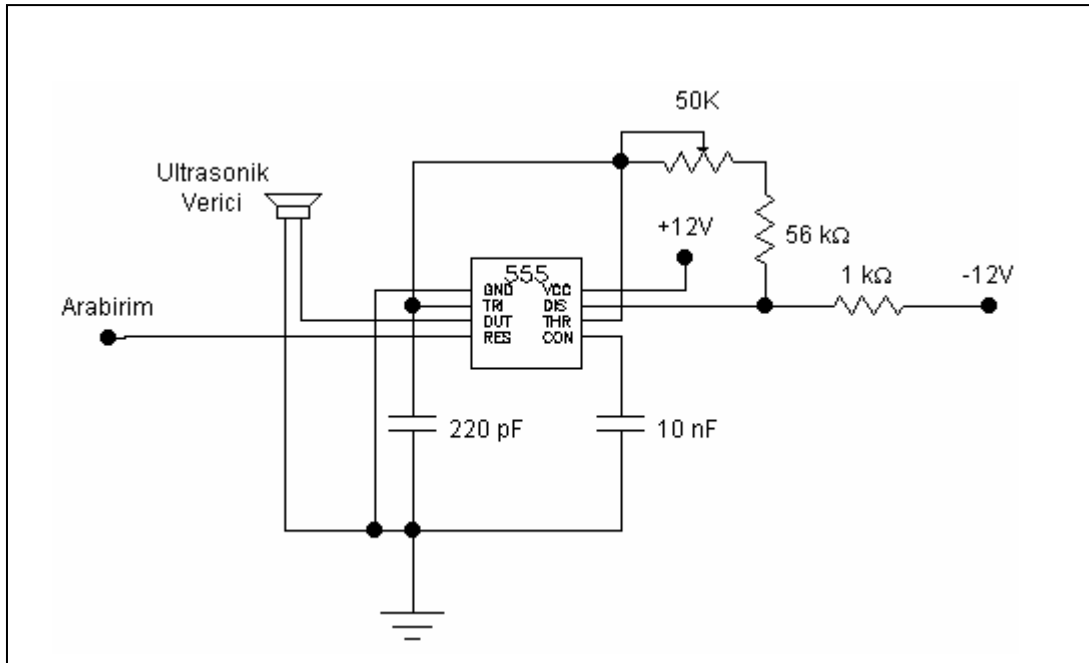
Burada düşünülmesi gereken nokta, her bir algılayıcının, arabirimden iki uç işgal edecektir. Bu 4 algılayıcı için 8 uç demektir. Bunun yerine algılayıcıların alıcı kısmı ortak bir uca bağlanabilir. Herhangi biri sesi algıladığında bu uç aktif olacaktır. Vericilerin farklı olduğu, ve bir algılayıcının sadece kendi vericisinden yayılan sesi algıladığı düşünülürse, her hangi birinin aktif olması demek, ona ait vericiden yayılan sesin geri geldiğini gösterir. Zaten bir alıcı, diğer bir algılayıcının vericisinden yayılan sesi algılasa bile, bu vericinin kendi alıcısından çok daha sonra olacağı için bu otomatikman işleme katılmayacaktır. Dolayısıyla 4 algılayıcı için 8 uç yerine 4 verici, 1 de alıcılar için toplam 5 uç yeterli olacaktır. Ultrasonik algılayıcı devrelerinin arabirime bağlantısı *şekil 6*'daki gibi olmalıdır. *Şekil 6*'da elemanların besleme ve toprak girişleri gösterilmemiştir. Verici ve alıcı devreler üzerinde ayrıntıya, çalışma durumu dışında girmeyeceğim. Verici girişi aktif (set) yapıldığı anda ultrasonik ses dalgası yayılmaya başlar. Ve bu giriş aktif olduğu sürece bu devam eder. Alıcı çıkışı normalde düşük seviyededir. Ses algılandığı anda ve algılandığı müddetçe bu çıkış yüksek seviye olacaktır. Piyasada sırf bu iş için üretilmiş ve bu mantıkla çalışan çok çeşitli modüller

mevcut. Giriş, çıkış ve besleme uyumluluğu dışında herhangi bir ayar yapmaksızın bu sistemde kullanılabilir.



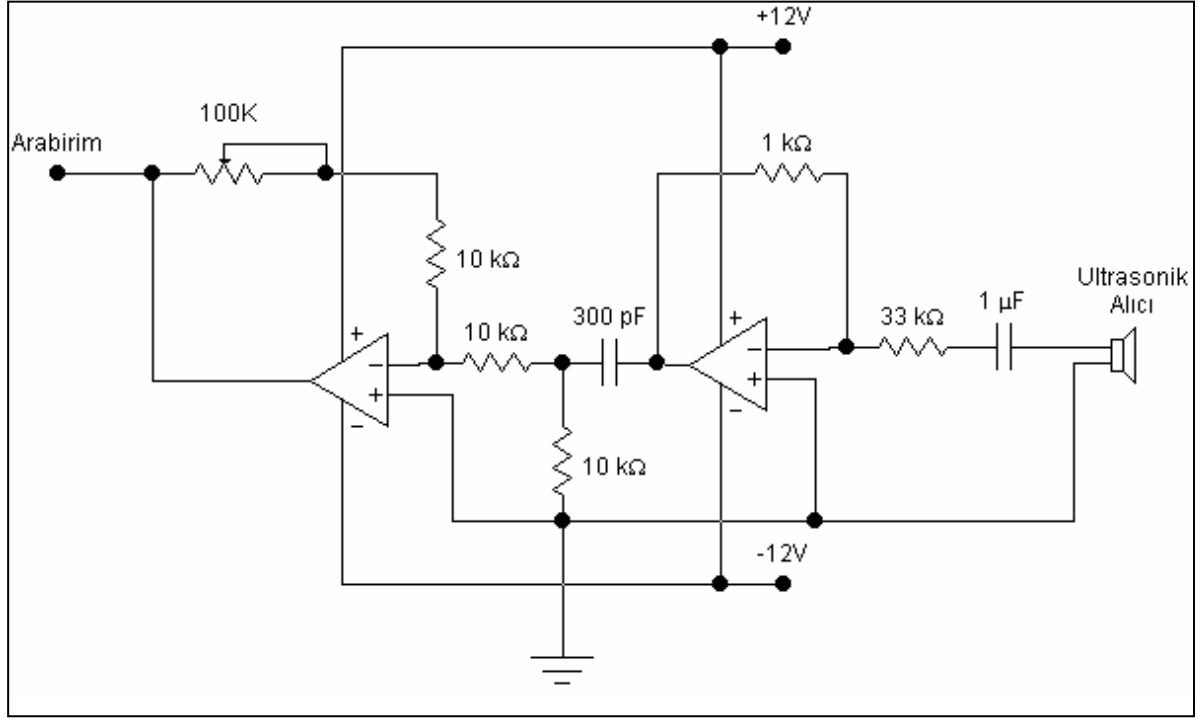
Şekil 6: Ultrasonik algılayıcıların arabirim bağlantısı

Şekil 7 ve şekil 8'de verici ve alıcı kısma ait devre şemaları verilmiştir.



Şekil 7: Ultrasonik verici devresi





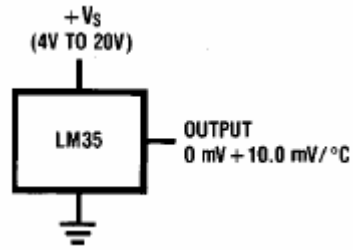
Şekil 8: Ultrasonik alıcı devresi

## A.2. Sıcaklık sensörü

Sistemde LM 35 entegresi kullanıldı.

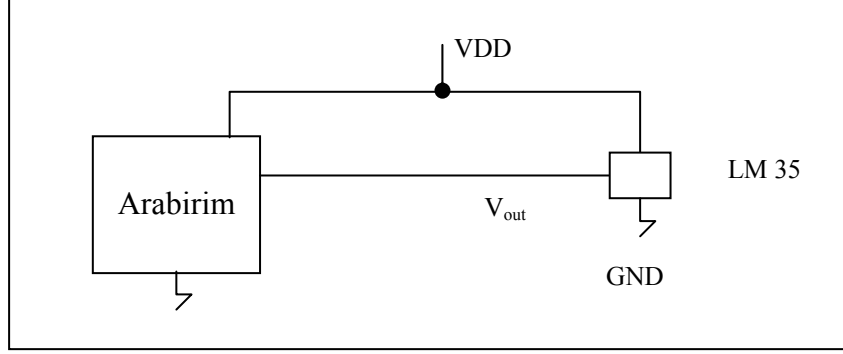


Şekil 7: LM35 bacak bağlantıları



Şekil 8: LM35 tipik uygulaması

0 ile 85 derece arasındaki sıcaklıklarda değişen her dereceye karşılık 10 mV'luk çıkış göstermekte. Herhangi bir kalibrasyona ihtiyaç duymaması (0 derece= 0 Volt), 4-20 V geniş besleme aralığına sahip olması ve basit bağlantı özellikleri LM 35'i bu sistem için en uygun sıcaklık sensörü haline getiriyor. Şekil 7'de bacak bağlantıları, şekil 8'de en yaygın kullanımı şekil 9'da ise arabirime bağlantısı görülmekte.



Şekil 9: LM 35 arabirim bağlantısı

Sıcaklık değerleri sonuçta bilgisayara aktarılacağından bir analog dijital çeviriciye ihtiyacımız olacak. Bunu da arabirim içerisinde gerçekleştirmek durumundayız. Bununla birlikte seviye tespiti için hassas zaman sayıcısı gerekmekte. Sistemde bunu da arabirim gerçekleştiriyor. Bilgisayara aktarılan bilgi eşdeğer dijital sinyallerden ibaret olacak. Aslında seviye tespitini de bilgisayar üzerinden yapmak mümkün ancak arabirim ile haberleşmesi ve arabirimin ilgili yüksek seviye bilgisini göndermesi için geçen süre bilginin doğruluğunu bu noktada direkt olarak etkileyeceğinden optimum çözüm olmayacaktır.

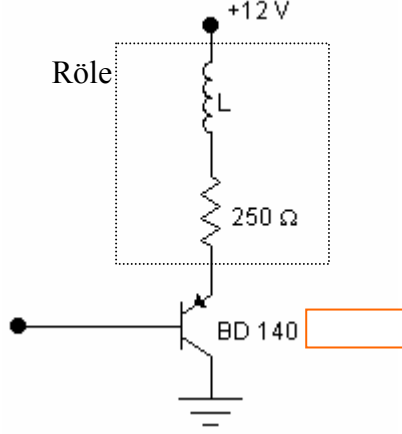
### B. Sürücü Devresi

Sistemde kontrol edilmesi gereken yüksek enerji seviyesinde elemanlar mevcut. Dolayısıyla TTL seviyesindeki bir sinyal bunları sürmeye yeterli olmayacaktır. Bu problemi bir sürücü devre ile gidermemiz gerekir. Arabirimin TTL seviyesinde olduğunu farz ederek 220 V AC-DC, 1-10 A'lık anahtarlama kapasitesi elde etmemiz yeterli olacaktır. Burada anahtarlama kelimesini kullanmamın sebebi frekans anahtarlama kastetmek değil, devrenin satrasyonda çalışmasını belirtmek içindir. Sürücü devrenin çıkışı ne kadar büyük olursa giriş için gerekli sinyal de o kadar büyük olacaktır. Aksi halde devre çok pahalı olur. Yüksek enerji seviyesinden kademe kademe aşağı inerek TTL seviyesine erişmeye çalışalım. Öncelikle sürücü sistemlerin beklenen özelliklerini sıralayalım. (yada anahtar devre)

- 1- hızlı olmalı
- 2- iletimdeyken gerilim düşümü az olmalı
- 3- kesimde iken sızıntı akımı az olmalı
- 4- elektriksel izolasyon sağlamalı vs.

Sürücü düzeneğimiz kademeli olacaktır. Elektriksel izolasyonu en sona koyarsak devremizi yüksek güçlü akım ve gerilimlere karşı korumuş oluruz. Bunu da röle ve kontaktörlerle sağlayabiliriz. En sonda kullanacağımız 380 V, 30 A lik bir kontaktörü piyasada çokça bulunan 12 V DC ile çalışan herhangi düşük güçlü bir röle ile sürebiliriz. O yüzden kontaktör için ayrıntıya girmeyeceğim. 12 V düşük güçlü bir röle tipik sürme direnci 200-300  $\Omega$  civarındadır. Bu da 40-60 mA lik bir çekme akımına karşılık gelir. Bu akımı BD 140

transistörü ile soğutucu kullanmadan rahatlıkla sağlayabiliriz. *Tablo 2*'de BD 140'ın katalog bilgileri görülmektedir.



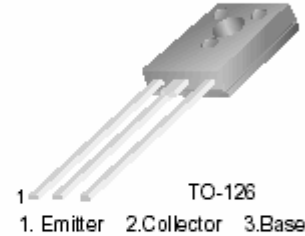
$V_{CE(sat)}$  0.5 V için röleye düşen gerilim 11.5 V olmaktadır. Ancak bu röle için problem değildir. 9 Volta kadar röle işlevini sürdürür. *Şekil 10*'da görülen bağlantı şekli için  $I_C=46$  mA ve  $I_B=1.17$  mA olmaktadır. BD 140 pnp olduğundan base akımı arabirime doğrudur. Arabirimden yüksek seviye sinyal geldiğinde çalışmasını sağlarsak, daha standart bir devre elde etmiş oluruz. BD 140'ın base'ini npn bir transistorün kolektörüne bağlarsak işlem çözülmüş olur.

*Şekil 10: BD 140 – Röle bağlantısı*

## BD136/138/140

### Medium Power Linear and Switching Applications

- Complement to BD135, BD137 and BD139 respectively



### PNP Epitaxial Silicon Transistor

#### Absolute Maximum Ratings $T_C=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
$V_{CB0}$	Collector-Base Voltage : BD136	- 45	V
	: BD138	- 60	V
	: BD140	- 80	V
$V_{CE0}$	Collector-Emitter Voltage : BD136	- 45	V
	: BD138	- 60	V
	: BD140	- 80	V
$V_{EB0}$	Emitter-Base Voltage	- 5	V
$I_C$	Collector Current (DC)	- 1.5	A
$I_{CP}$	Collector Current (Pulse)	- 3.0	A
$I_B$	Base Current	- 0.5	A
$P_C$	Collector Dissipation ( $T_C=25^\circ\text{C}$ )	12.5	W
$P_C$	Collector Dissipation ( $T_a=25^\circ\text{C}$ )	1.25	W
$T_J$	Junction Temperature	150	$^\circ\text{C}$
$T_{STG}$	Storage Temperature	- 55 – 150	$^\circ\text{C}$

#### Electrical Characteristics $T_C=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
$V_{CE0(sus)}$	* Collector-Emitter Sustaining Voltage	$I_C = - 30\text{mA}, I_B = 0$	- 45			V
	: BD136		- 60			V
	: BD138		- 80			V
	: BD140					
$I_{CB0}$	Collector Cut-off Current	$V_{CB} = - 30\text{V}, I_E = 0$			- 0.1	$\mu\text{A}$
$I_{EB0}$	Emitter Cut-off Current	$V_{EB} = - 5\text{V}, I_C = 0$			- 10	$\mu\text{A}$
$h_{FE1}$	* DC Current Gain	$V_{CE} = - 2\text{V}, I_C = - 5\text{mA}$	25			
$h_{FE2}$		$V_{CE} = - 2\text{V}, I_C = - 0.5\text{A}$	25			
$h_{FE3}$		$V_{CE} = - 2\text{V}, I_C = - 150\text{mA}$	40		250	
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = - 500\text{mA}, I_B = - 50\text{mA}$			- 0.5	V
$V_{BE(on)}$	* Base-Emitter ON Voltage	$V_{CE} = - 2\text{V}, I_C = - 0.5\text{A}$			- 1	V

\* Pulse Test: PW=350 $\mu\text{s}$ , duty Cycle=2% Pulsed

#### $h_{FE}$ Classification

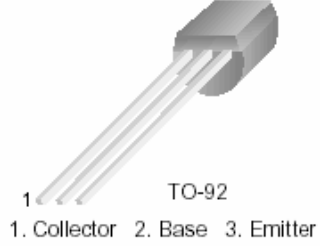
Classification	6	10	16
$h_{FE3}$	40 – 100	63 – 160	100 – 250

Tablo 2: BD 140 katalog bilgileri

## BC237/238/239

### Switching and Amplifier Applications

- Low Noise: BC239



### NPN Epitaxial Silicon Transistor

#### Absolute Maximum Ratings $T_a=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units	
$V_{CES}$	Collector-Emitter Voltage	: BC237	50	V
		: BC238/239	30	V
$V_{CEO}$	Collector-Emitter Voltage	: BC237	45	V
		: BC238/239	25	V
$V_{EBO}$	Emitter-Base Voltage	: BC237	6	V
		: BC238/239	5	V
$I_C$	Collector Current (DC)	100	mA	
$P_C$	Collector Power Dissipation	500	mW	
$T_J$	Junction Temperature	150	$^\circ\text{C}$	
$T_{STG}$	Storage Temperature	-55 ~ 150	$^\circ\text{C}$	

#### Electrical Characteristics $T_a=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
$BV_{CEO}$	Collector-Emitter Breakdown Voltage	$I_C=2\text{mA}, I_B=0$	: BC237	45		V
			: BC238/239	25		V
$BV_{EBO}$	Emitter Base Breakdown Voltage	$I_E=1\mu\text{A}, I_C=0$	: BC237	6		V
			: BC238/239	5		V
$I_{CES}$	Collector Cut-off Current	$V_{CE}=50\text{V}, V_{BE}=0$ $V_{CE}=30\text{V}, V_{BE}=0$		0.2	15	nA
			: BC238/239		0.2	15
$h_{FE}$	DC Current Gain	$V_{CE}=5\text{V}, I_C=2\text{mA}$	120		800	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C=10\text{mA}, I_B=0.5\text{mA}$		0.07	0.2	V
		$I_C=100\text{mA}, I_B=5\text{mA}$		0.2	0.6	V
$V_{BE(sat)}$	Collector-Base Saturation Voltage	$I_C=10\text{mA}, I_B=0.5\text{mA}$		0.73	0.83	V
		$I_C=100\text{mA}, I_B=5\text{mA}$		0.87	1.05	V
$V_{BE(on)}$	Base-Emitter On Voltage	$V_{CE}=5\text{V}, I_C=2\text{mA}$	0.55	0.62	0.7	V
$f_T$	Current Gain Bandwidth Product	$V_{CE}=3\text{V}, I_C=0.5\text{mA}, f=100\text{MHz}$		85		MHz
		$V_{CE}=5\text{V}, I_C=10\text{mA}, f=100\text{MHz}$	150	250		MHz
$C_{cb}$	Output Capacitance	$V_{CB}=10\text{V}, I_E=0, f=1\text{MHz}$		3.5	6	pF
$C_{ib}$	Input Base Capacitance	$V_{EB}=0.5\text{V}, I_C=0, f=1\text{MHz}$		8		pF
NF	Noise Figure	$V_{CE}=5\text{V}, I_C=0.2\text{mA},$ $f=1\text{KHz}, R_G=2\text{K}\Omega$		2	10	dB
		: BC239			4	dB
		: BC239	$V_{CE}=5\text{V}, I_C=0.2\text{mA}$ $R_G=2\text{K}\Omega, f=30\sim 15\text{KHz}$			4

#### $h_{FE}$ Classification

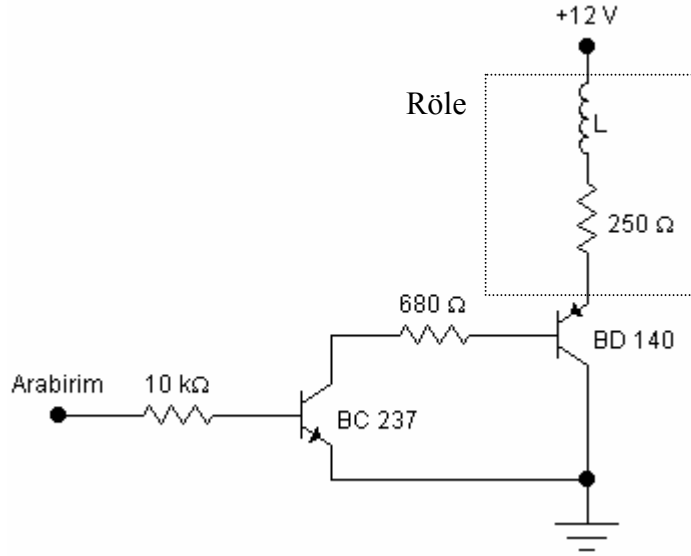
Classification	A	B	C
$h_{FE}$	120 ~ 220	180 ~ 460	380 ~ 800

Tablo 3: BC 237 katalog bilgileri

BC 237 bu iş için uygundur. *Tablo 3*'te BC 237'nin katalog bilgileri yer almaktadır.  $V_{CE(sat)}=0.2$  V için BD 140'ın  $V_{BE}$ 'ine yaklaşık 0.8 V gibi bir gerilime daha ihtiyaç duyulacaktır. Bu koldaki akım 1.17 mA idi. Dolayısıyla

$$R = 0.8 / 0.00117$$

yaklaşık 680  $\Omega$  bulunur. BC 237'yi 5 V TTL seviyesi ile biyashıyacağımızı düşünürsek girişe koyacağımız 10k $\Omega$ 'luk bir direnç için betasının 2 den fazla olması yetecektir ki öyledir de zaten. Devre bu düşünceden yola çıkarak *şekil 11*'deki son halini alır. Ve arabirimden en fazla 0.42 mA akım çeker.



*Şekil 11: Sürme devresi*

Devrede sürülen röle ile 5 A, 240V AC yada 5 A, 30 V DC yükler tahrik edilebilir. Devre bu hali ile soğutucuya gerek duymaz çünkü anahtarlama elemanları katalog sınırlarının çok çok altında çalışmaktadır. TTL seviyesinde 0.42 mA gibi çok küçük bir akım çekerek, sürücü entegreyi yormayacaktır. Sıvı seviye kontrolü, sıcaklık kontrolü gibi gecikme içeren on-off kontrol sistemleri için ekstra gecikme sadece röle kontaklarının hareketinde meydana gelir ki bu gecikme bile bu kontrol sistemleri için ihmal edilebilecek kadar azdır. Zira transistörlerin çalışma frekansı MHz'ler mertebesindedir.

Sistemde şimdiye kadar anlatılanlar donanımın arabirime kadar olan kısmıydı.

### C. Arabirim

Sistemdeki işin büyük bir kısmı arabirim tarafından gerçekleştirilmektedir. Seviye tespiti için gereken hassas zamanlama, sıcaklık bilgisi için gereken analog dijital çevrici, sürme uçları ve PC haberleşmesi için gereken donanım arabirim devresi içerisinde yer almaktadır. Bunlara ek olarak, içerisinde haberleşme trafiğinin sürekliliğini sağlamak için küçük çapta hafıza modülleri de bulunmalıdır. Bahsedilen özelliklere sahip bir devre tasarlamak hem analog hem de sayısal açıdan mümkündür ve sayısız kombinasyona sahiptir. Mikroişlemci omurgalı bir kart akla ilk gelen model olmakla birlikte, sayılan diğer komponentleri de tasarlamak gerekir. (ADC, PC haberleşme arabirimi vb.) Bütün bunları bünyesinde barındıran kapalı devre programlanabilen entegreler de mevcuttur. En bilineni Microchip firmasının ürettiği PIC mikroişlemcileridir.

Çok çeşitli konfigürasyonlarda üretilen Pic mikroişlemcileri sahip olduğu özellikler sayesinde TTL seviyesindeki hemen hemen tüm elektronik uygulama için akla gelen ilk yapı olmaktadır. Klasik mikroişlemci kartlarındaki tüm donanımı ve fazlasını bünyesinde bulundurmasının yanında düşük güç tüketmesi özelliği ile de özellikle küçük boyutlu ve taşınabilir devreler için en uygun model olmaktadır. Ben de sistemdeki arabirim devresini PIC16F877 işlemcisi üzerine kurdum. Bununla ilgili ayrıntıyı sonraki bölümlerde izah etmeye çalışacağım.

### C.1. Haberleşme Standardı Seçimi

Bilgisayarlar temel olarak paralel ve seri olmak üzere iki tür haberleşme standardını kullanır. Her ikisinin de birbirine göre üstün yanları vardır. Paralel veri iletimi 8 bitlik veri yolu kullanır. Bu da her bir zaman çevriminde (clock puls) 256 ( $2^8$ ) karakterden bir tanesinin gönderilmesini olanaklı kılar. Seri veri iletiminin veri yolu tek bitlidir. Bu tek bit, zaman üzerinde paylaşılarak her bir zaman çevriminde bir karakterin yalnız bir bitinin gönderilmesini olanaklı kılar. Bu paralel haberleşmeyi üstün kılıyor gibi gözükmesine rağmen, daha az veri yolu kullanmak bir mühendislik düşüncesi gereğidir. Her iki haberleşme standardını kıyaslamak için tablo 4'e göz atalım:

<i>Paralel iletim</i>	<i>Seri iletim</i>
Çok hızlı bilgi gönderilip – alınabilir	Paralel kadar hızlı değildir
Kullanılan kablo uzunluğu 9 feet olabilir.	Kullanılan kablo uzunluğu kısıtlanmaz.
Paralel iletim kabloları çokludur.	Seri iletim kabloları genelde çifttir.
Bilginin bir karakteri bir kerede gönderebilir.	Karakterin bir biti bir kerede gönderilir.
Senkronizasyon için bir hat işgal eder	Asenkron çalışmayı destekler

Tablo 4: Paralel ve seri haberleşme standartları karşılaştırması

Endüstrideki bir çok uygulama için seri haberleşme yeterli, çoğu zaman da gereklidir. Bu uygulama için de seri haberleşme hızı yeterli olmaktadır. Bu yüzden seri haberleşme standardını seçtik.

### C.2. Asenkron Seri Haberleşme Standardı

Bir karakterin bitleri tek bir hat üzerinden sırayla gönderilir. Karakterin başında gönderilen başlama bitinden sonra alıcı ve verici birbirinden bağımsızdır. Verici, bitleri belirlenmiş bir hızda gönderirken, alıcı da teorik olarak aynı hızda bunları okur. Asenkron seri haberleşmenin önemli terimlerini şu şekilde sıralayabiliriz:

**Baud Rate:** Veri iletim hızıdır. 1 saniyede iletilen bit adedine denir. Standart olarak veri hızları 300,600,1200,2400,4800,9600,19200, ... şeklindedir.

**Start Bit:** Seri haberleşmede veriler senkron yada asenkron olarak iletilebilirler. Senkron iletimde belirli bir başlangıçtan sonra veriler sıralı olarak ard arda gelirler. Uzun bir dosyanın iletiminde bu yol kullanılabilir. Asenkron iletimde ise bir veri gönderildikten sonra yeni bir veri belirli olmayan bir zamanda gelir. İşte bu yeni verinin başlangıcı start biti ile bildirilir.

**Stop Bit:** Gönderilen verinin bittiğini ifade eden bittir. Bu biti alan alıcı yeni bir veri için start bitini gözlemeye başlar. haberleşmenin tipine göre 1 yada 2 bit uzunluğunda olabilir.

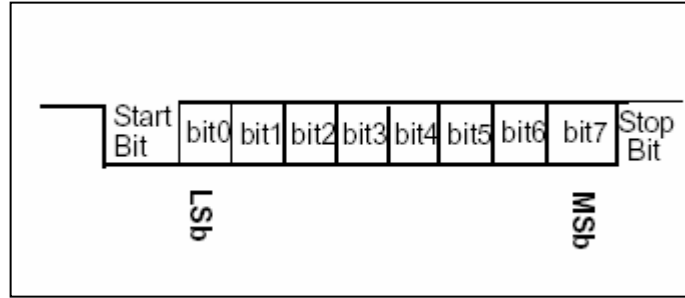
**Eşlik Biti (Parity Bit):** Bir çeşit hata denetim bitidir.start ve 8 bitlik bir veri iletildikten sonra stop biti gönderilmeden parity gönderilir. iletilen 8 bitlik veride 1 lerin sayısı çift ise EVEN tek ise ODD biti gönderilir. Alıcı bu biti kontrol ederek alınan verinin doğru olup olmadığını kontrol eder. 1 parity 1 bit uzunluğundadır.

**Siplex:** Bilginin sadece bir yönde aktığı durumdur.

**Half duplex:** Bilginin farklı zamanlarda çift yönde akabildiği durumdur.

**Full duplex:** Bilginin aynı anda çift yönde akabildiği durumdur.

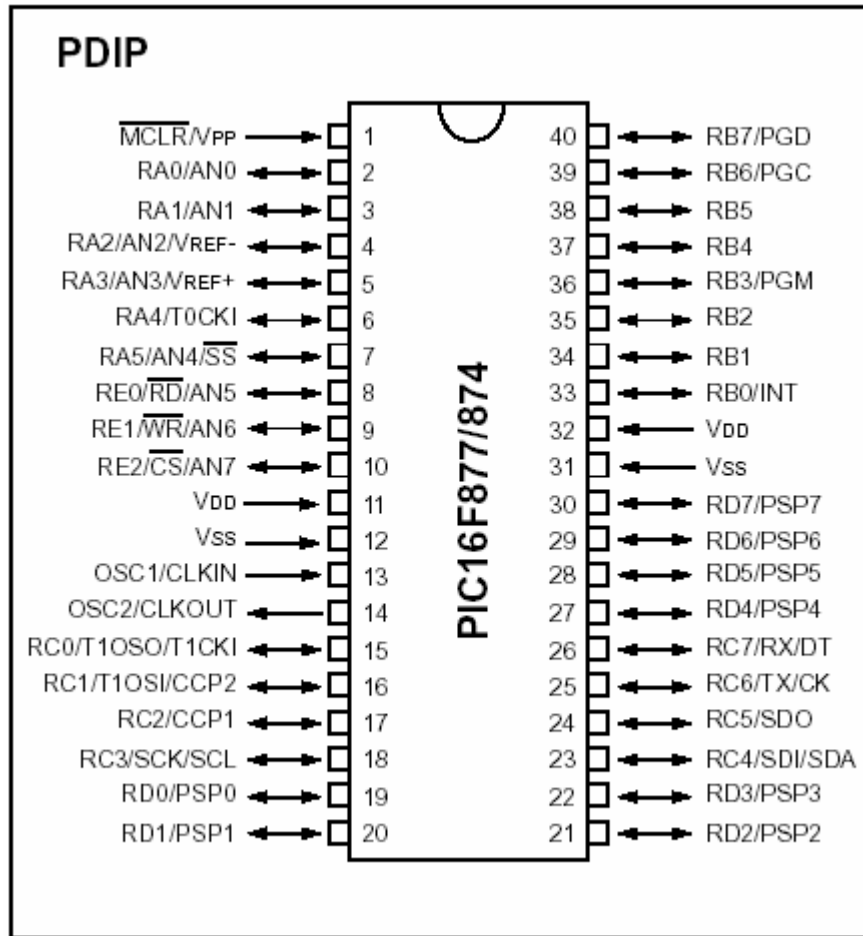
Şekil 12’de asenkron seri haberleşme zaman diyagramı görülmektedir.



Şekil 12: asenkron seri haberleşme zaman diyagramı

### C.3. PIC16F877 Mikroişlemcisi

40 pin, 8 bit, flash tip mikro denetleyicidir. Bünyesinde USART, 8 adet 10 bit ADC, 3 adet birbirinden bağımsız zamanlayıcı, 14 farklı kesme vektörü, 16 bit karşılaştırıcı, 16 bit tutucu, 10 bit PWM, 8 bit harici paralel portu barındırır. Her birine farklı görevler atanabilen 33 giriş-çıkış uç içeren denetleyici toplam 40 pindir. Şekil 13’te pin diyagramı görülmektedir.



Şekil 13: PIC16F877 pin diyagramı

Kullanılacak özelliklerden usart ve adc’yi açıklamaya çalışalım...



#### C.4. PIC16F877 Usart

PIC16F877’de seri port olarak veri göndermek (transmit) için RC6 pini, veri almak (receive) için RC7 pini kullanılmakta. Bu pinlerin statülerini ve tüm seri haberleşmeyi kontrol eden 2 yazmaç var. Seri haberleşme yapmak için öncelikle bu pinlerin RX ve TX olarak ayarlanması gerekiyor. Sonrasında yine haberleşme ile ilgili bilgilerin (baud hızı, stop bit sayısı, parity durumu) bu 2 yazmaca yazılması gerekiyor.

REGISTER 10-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

Bit 7: CSRC :Zamanlamanın kaynağını belirliyor. Asenkron haberleşmede değeri önemli değil (0-1)

Bit 6: TX9 :9 bit veri gönderme izin biti.

1 : 9 bit gönderim seçer

0 : 8 bit gönderim seçer

Bit 5: TXEN :Transmit izin biti.

1 : gönder

0 : gönderme

Bit 4: SYNC :Usart mod seçme biti

1 : senkron mod

0 : asenkron mod

Bit 3: - : Kullanılmıyor. Okunduğunda sıfır değeri veriyor.

Bit 2: BRGH :Hızlı-yavaş baud hızı seçme biti

1 : yüksek hız

0 : düşük hız

Bit 1: TRMT :Transmit kaydırma yazmacı durum biti (shift register)

1 : TRM boş

0 : TRM dolu

Bit 0 : TX9D : 9 bit haberleşmenin 9. biti. Parity biti de olabilir.

**REGISTER 10-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

Bit 7: SPEN :Seri port seçme biti  
1 : RC7 ve RC6 pinlerini seri port için ayarlar  
0 : Seri port ayarlaması yok

Bit 6: RX9 :9 bit veri alma izin biti.  
1 : 9 bit alma seçer  
0 : 8 bit alma seçer

Bit 5: SREN :Tek data alma izin biti. Asenkron moda kullanılmıyor

Bit 4: CREN :Devamlı alma izin biti  
1 : devamlı alma aktif  
0 : devamlı alma pasif

Bit 3: ADDEN :RX9 biti 1 iken adres sorgusu yapmak için kullanılıyor. Bu sistemde kullanılmayacak.

Bit 2: FERR :Veri kayma hata biti (framing error)  
1 : kayma hatası var  
0 : kayma hatası yok

Bit 1: OERR :Taşma hata biti (overrun)  
1 : taşma var  
0 : taşma yok

Bit 0 : RX9D : 9 bit haberleşmenin 9. biti. Parity biti de olabilir.

İşlemci yazılımının başında bu registerlere kullanılacak konfigürasyona göre gereken değerler yazılacaktır. Baud hızı için ise SPBRG registeri kullanılıyor. Buraya yazılan 8 bitlik değer ile baud hızı *tablo 5*'teki gibi yapılıyor.

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{OSC}/(64(X+1))$	Baud Rate = $F_{OSC}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{OSC}/(4(X+1))$	N/A

*Tablo 5: baud hızı formülasyonu*

Burada  $F_{OSC}$  kullanılan osilatör frekansını, X ise SPBRG registerine yazılması gereken değeri temsil ediyor. Elbetteki maksimum 20 MHz saat frekansında çalışan bir denetleyicinin bütün baud hızlarında, X değerlerinin tam doğru değerler vermesi mümkün değildir. *Tablo 6* ve *tablo 7*'de BRGH'ın 0 ve 1 olduğu durumlar için, çeşitli saat frekanslarında ve baud hızlarındaki bağlı hata görülmektedir.

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.68	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

Tablo 6: BRGH=0 için bağıl hata değerleri

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000	-	0	625.000	-	0

Tablo 7: BRGH=1 için bağıl hata değerleri

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

Burada bizim ilgilendiğimiz frekans 3.6864 MHz dir. Baud hızını ise 9600 seçiyoruz. Bu şekilde teknik olarak bağıl haberleşme hatasını sıfır kabul ediyoruz..

## C.5. PIC16F877 ADC

Analog dijital çevirici modülü kullanmak için ADCON0 ve ADCON1 registerlerini işleme dahil etmemiz gerekiyor. İlgili ayarlar bu registerlerden yapılıyor. 10 bit için 8 bitlik 2 register kullanılıyor. ADRESH ve ADRESL registerleri çevrim değerini tutuyor.

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

Bit 7-6: ADCS1, ADCS2 :ADC saati seçme biti

00 :  $F_{OSC}/2$

01 :  $F_{OSC}/8$

10 :  $F_{OSC}/32$

11 :  $F_{RC}$  dahili RC osilatör

Bit 5-3: CHS2,CHS1,CHS0 :Analog kanal seçme biti

000 : kanal 0, (RA0)

001 : kanal 1, (RA1)

010 : kanal 2, (RA2)

011 : kanal 3, (RA3)

100 : kanal 4, (RA5)

101 : kanal 5, (RE0)

110 : kanal 6, (RE1)

111 : kanal 7, (RE2)

Bit 2:  $\overline{GO/DONE}$  :A/D çevrim durum biti

1 : çevrimi başlatmak için 1 yapılır, çevrim tamamlanınca otomatik olarak 0 olur.

0 : ADC meşgul değil

Bit 1: - :Kullanılmıyor. 0 okunur.

Bit 0: ADON :A/D modül açık biti

1 : modül açık, işlem sürüyor

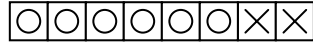
0 : modül kapalı, çevrim akımı yok

**REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)**

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Bit 7: ADFM :A/D sonuç formatı

1 : sağa dayalı sonuç formatı, ADRESH'in en önemli 6 biti 0 okunur

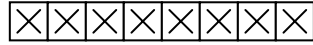


ADRESH

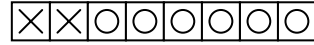


ADRESL

0 : sola dayalı sonuç formatı, ADRESL'nin en önemsiz 6 biti 0 okunur



ADRESH



ADRESL

Bit 6-4: - :Kullanılmıyor. 0 okunuyor.

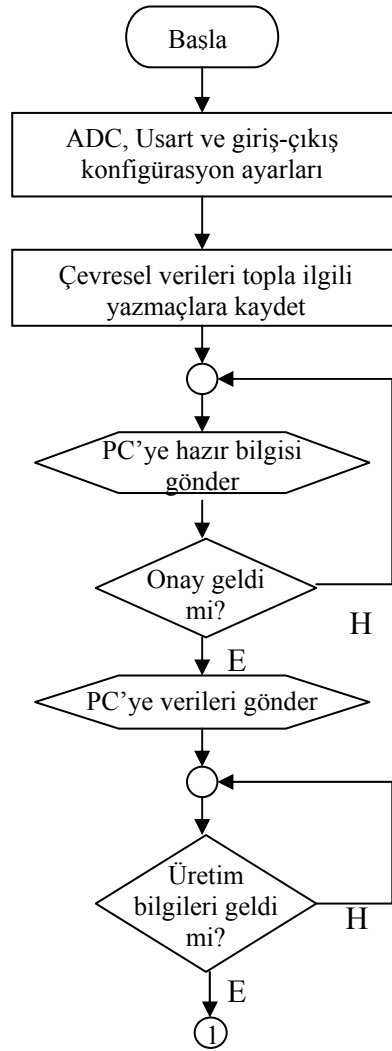
Bit 3-0: PCFG3, PCFG2, PCFG1, PCFG0 : A/D port konfigürasyon kontrol bitleri

PCFG3: PCFG0	AN7 <sup>(1)</sup> RE2	AN6 <sup>(1)</sup> RE1	AN5 <sup>(1)</sup> RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs <sup>(2)</sup>
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011X	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

Bu sistemde sadece 1 tane A/D kanal gerektiğinden 1110 seçmemiz yeterli olacaktır. Konfigürasyon seçimi ilerde anlatılacaktır.

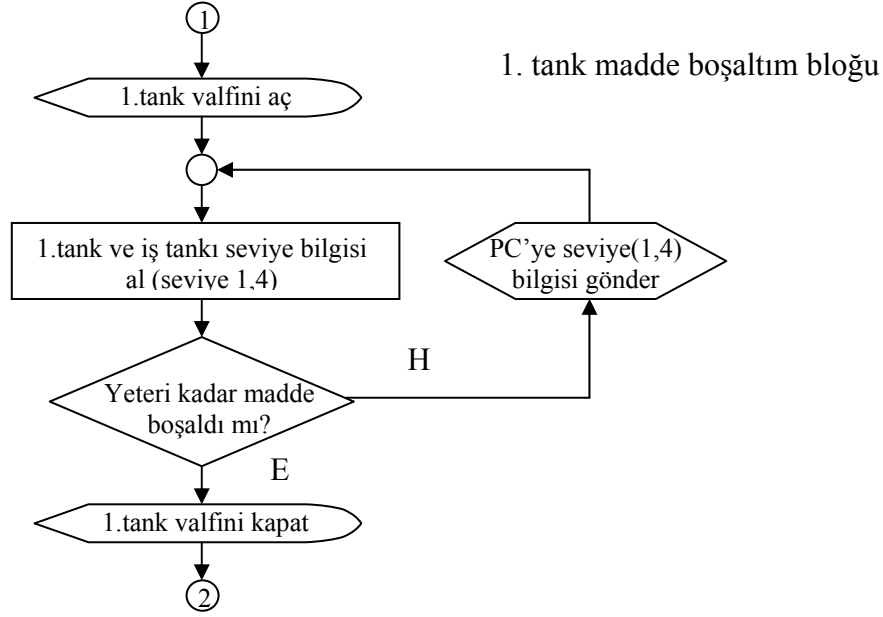
### C.5. Arabirim Yazılım Akış Şeması

PIC16F877'nin buradaki temel işlevi, PC ile üretim sistemi arasında bir tercüman gibi davranarak, her iki tarafın da birbirleriyle haberleşebilmelerini sağlamaktır. Üretim sistemindeki verileri, uygun formata çevirip PC'ye aktarır, PC'den gelen verilere göre de ilgili uçları aktif veya pasif yapar. Bunun yanında, küçük karşılaştırma işlemlerini üzerine alarak PC'nin yükünü hafifletir. Bu şekilde PC yönetici, PIC uygulayıcı ve çevirmendir. PIC yaptığı her hareketi, aldığı her bilgiyi yöneticisine rapor eder. PIC16F877 için gereken yazılımın akış şemasını adım adım hazırlayalım. Şekil 14'te başlangıç hazırlığı gösterilmiştir.



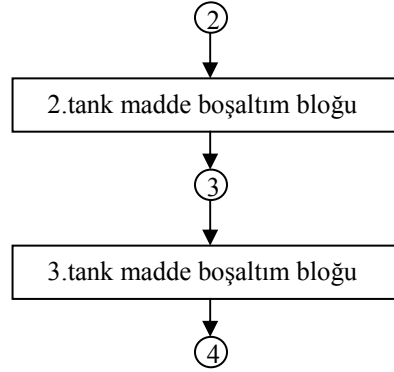
Şekil 14: Pic başlangıç akış şeması

Buraya kadar PIC, başlangıç için gereken ayarları yapmış, tanklardaki seviye ve sıcaklık bilgilerini almıştır. PC'ye hazır olduğunu bildirir. Bağlantı onayından sonra açılıştaki üretim tankları durumunu PC'ye ilk kez gönderir. Bu görsel simülasyon için gereklidir. Bu aşamaya kadar PIC kendi başına yapması gerekenleri bitirmiştir. PC'den üretim bilgilerini bekler. Üretim bilgileri sırasıyla 1., 2., 3. tankların yeni seviyeleri ve iş tankının olması gereken sıcaklığıdır.



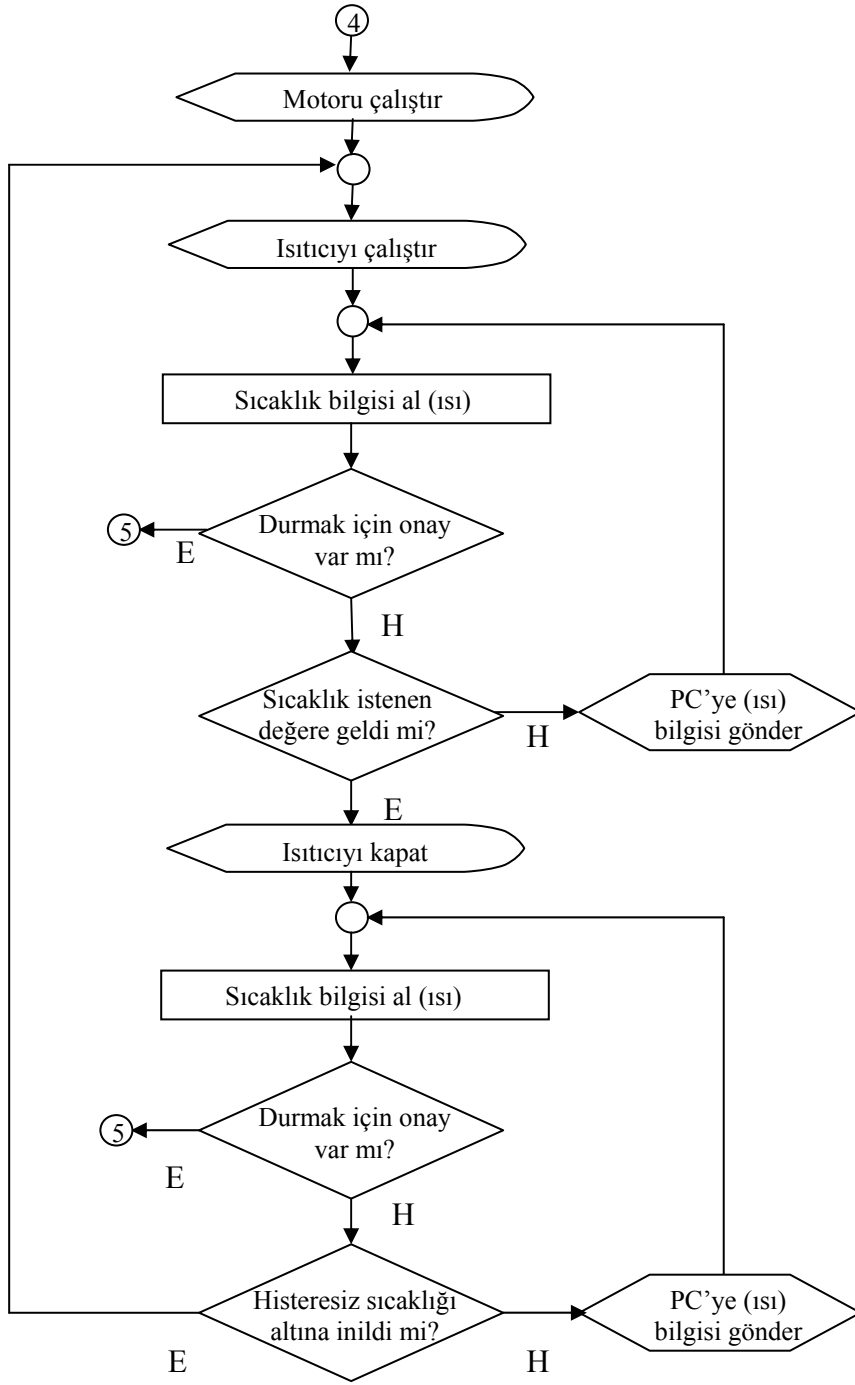
Şekil 15: 1.tank madde boşaltım yazılım akış şeması bloğu

Şekil 15'te üretim sürecinin ilk adımı olan, iş tankına madde alınmasına ait akış şeması görülmektedir. Bu bloğu "1.tank madde boşaltım bloğu" şeklinde adlandırırsak diğer tanklar için de indisler farklı olmak üzere aynı bloğu yazabiliriz.



Şekil 16: maddelerin iş tankına alınmasına ilişkin akış şeması bloğu

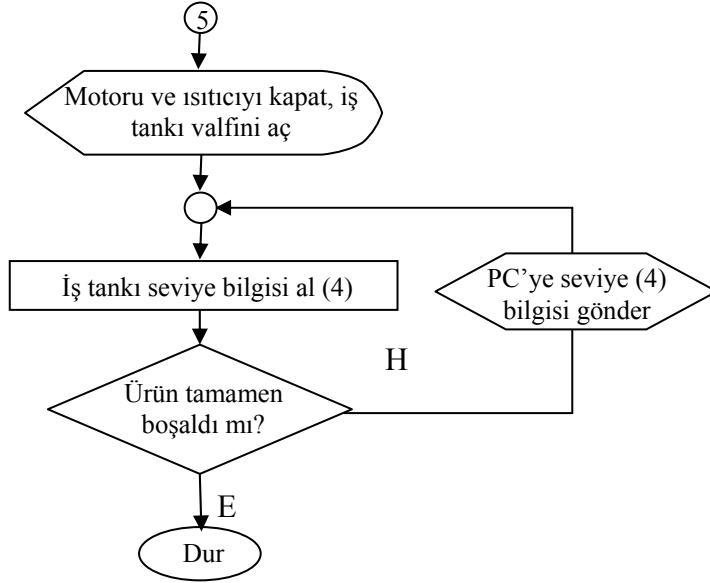
Bu blokları sırasıyla programa ilave edince, maddelerin sevki ile ilgili kısım bitmiş olur. (Şekil 16) Bundan sonra iş tankı ile ilgili işlemlere geçilir. Önce karıştırma motoru ve ısıtıcı açılır. Sıcaklık istenen seviyeye ilk geldiği anda (sıcaklık kontrolünde ayar noktasına defalarca ulaşılır) iş süresi saymaya başlar. Ve süre sonunda motor ve ısıtıcı kapatılarak, iş tankı valfi açılır. Ürün, ürün tankına alınır. Üretim tamamlanmıştır. Şekil 17'de buna ilişkin akış şeması bloğu görülmektedir. Süre sayma işini PC yazılımına bırakmak daha kolay olacaktır. Üretimin süresinin ne kadar olacağı belli değildir. Belki de saatler sürecektir. Ekstra bir zamanlama program bloğu yazmaktansa, PC üzerindeki hazır algoritmalar kullanılarak zamanlama daha kolay yapılır. Bunun için PC programının PIC den gelen sıcaklık bilgilerini test etmesi gerekir. Süreyi başlatmak için istenilen sıcaklığa gelmesini beklemelidir. Sıcaklık kontrolünü PIC yapacaktır. Ancak burada süreyi başlatan ve bitiren PC'dir. Bu yüzden PIC sıcaklık kontrolü sırasında sürenin bitip bitmediğini PC'ye sormalıdır. Yada bu esnada usart kesmesi aktif yapılabilir.



Şekil 17: üretim işlemine ilişkin akış şeması bloğu

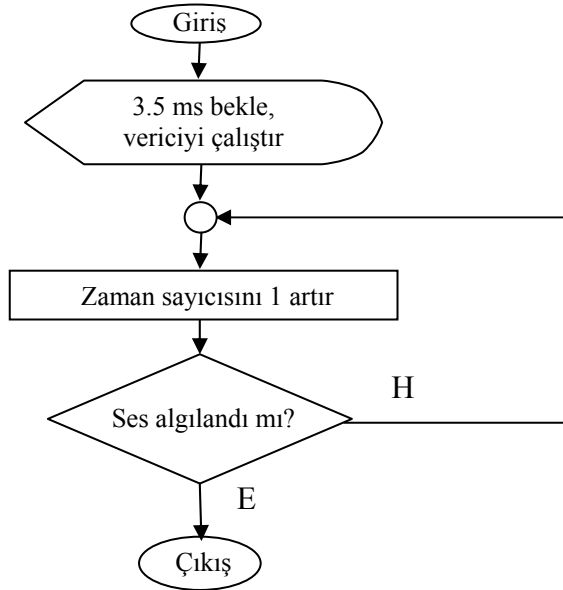
Program 5 nolu program parçasında son bulur. Motor ve ısıtıcı kapatılır. İş tankı valfi açılarak iş tankının ürün tankına boşaltılması sağlanır. Buna ilişkin akış şeması Şekil 18'de gösterilmiştir.





Şekil 18: program sonuna ilişkin akış şeması bloğu

Seviye bilgisi almak için gereken yazılımı bir alt program halinde tanımlarsak, akış şeması şekil 19'daki gibi olacaktır.



Şekil 19: seviye tespit alt programı akış şeması

## C.7. Pic-PC uyarlayıcı

PC seri iletişim için RS232 standardındaki COM portlarını kullanır. Seri iletişimi cazip kılan uzak mesafelere izin verebilmesi özelliği, farklı gerilim seviyeleri ile işlem yapmasından kaynaklanır ki bu da TTL seviyesindeki devreler ile arasında bir uyarlayıcıya ihtiyaç doğurur. RS232 standardında -25,-3 V arasındaki gerilim değerleri lojik 1, 3, 25 V arasında kalan gerilim değerleri de lojik 0'ı temsil eder. TTL de ise 5 V lojik 1, 0 V lojik 0'ı ifade eder. Burada pic ile PC arasında bu iş için üretilmiş MAX232 entegresini kullandım.

## D. PC Ortamı ve Arayüz Programı

PC ortamı operatör ile gerçek dünya arasında 1. iletişim modülü olması nedeniyle, ne kadar görsel ve işitsel malzeme içerirse verim artışı da o denli yüksek olur. Sürecin izlenmesi aşamasında basit bir komut satırı yerine, komutları içeren düğmeler, ikonlar, sistem ve alt sistemleri ifade eden bar grafikleri, etkileşimli resim dosyaları ve formlar, süreci temsil eden hareketli, ışıklı nesnelere kullanmak daha anlaşılır bir arayüz sunacaktır.

Sistemin bizden istediği parametreler şunlardır:

1. maddelerin karışım yüzdeleri veya miktarları
2. elde edilmek istenen ürünün miktarı (1.'de yüzde girilirse)
3. ürün hazırlanırken istenen sıcaklık
4. istenen sıcaklıkta karıştırma süresi

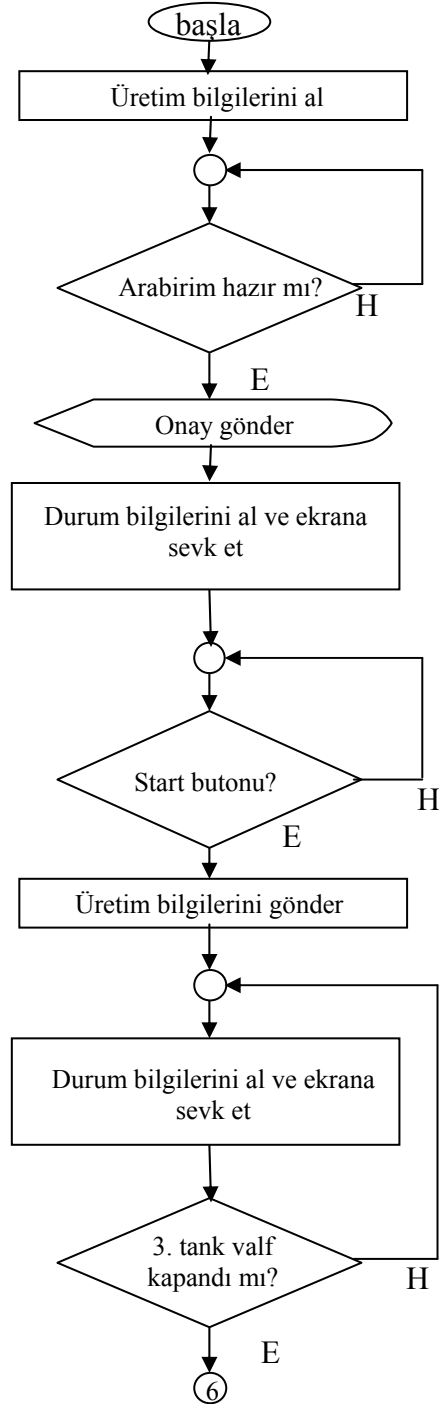
Sistemin bize her an bize sunacağı değerler ise şu şekilde:

1. hammaddelerin tanktaki miktarları
2. iş tankındaki madde miktarı
3. tüm valflerin durumu (açık-kapalı)
4. iş tankı sıcaklığı
5. karıştırma motoru durumu (çalışıyor-çalışmıyor)
6. üretim durumu (işleniyor-ürün hazır)

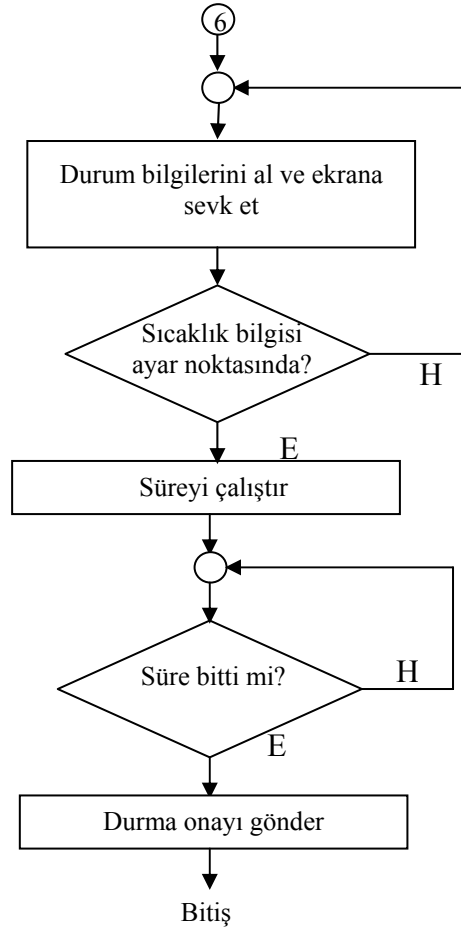
Dolayısı ile arayüz ekranında süreç esnasında 6 farklı değer giriş alanı (3 hammadde, 1 iş maddesi, sıcaklık ve süre), 11 farklı değer göstergesi (3 hammadde miktarı, 1 iş tankı miktarı, 4 valf durumu, sıcaklık, motor durum, süreç durum) bulunmalıdır. Arayüz programı Delphi ile hazırlanacaktır. Delphi hakkında ayrıntıya girmeyeceğiz ancak kısaca bahsetmekte fayda var: Delphi nesne tabanlı görsel bir programlama dilidir. Diğer alt seviye programlardan ayrılan en önemli özelliği, bir iş yapılacağı zaman ilgili öğenin aktif olmasıdır. Program kodları, programda yer alan nesnelereki değişimler içerisine yazılır. Bu klasik tabirle kesme ile çalışan programlara benzer ancak kesme olmadığı durumda bile herhangi bir döngüde çalışması gerekmez. Normalde bekleme modunda olan program bu esnada içerisinde kod yazılı nesne değişimlerini takip eder. Bu olay geri planda gerçekleşir. Örneğin, bir seri bilgi gelişinde yapılacak işlem için ilgili kodun port ile ilgili nesnenin “yeni bilgi geldi” olayına yazılması yeterlidir. Yeni bilgi geldiğinde program buraya yazılan kodu işleme sokar.

## D.1. Arayüz Program Akış Şeması

Delphi için kod yazarken klasik akış şemasını uyarlamamız gerekmektedir. Örneğin akış şemasında bulunan bazı döngüler delphi'de kullanılmayacaktır. Arayüz programı akış şeması *şekil 20*'deki gibi olmalıdır.



Bu işlem programın her aşamasında olabilmektedir...



Şekil 20: arayüz programı akış şeması