

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

## **AKVARYUM OTOMASYONU TASARIMI VE YAPIMI**

### **YÜKSEK LİSANS TEZİ**

**Faruk AKYILDIZ**

**Enstitü Anabilim Dalı : Elektronik ve Bilgisayar Eğitimi**

**Tez Danışmanı : Yrd. Doç. Dr. Ali Fuat BOZ**

**Mayıs 2007**

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

## **AKVARYUM OTOMASYONU TASARIMI VE YAPIMI**

### **YÜKSEK LİSANS TEZİ**

**Faruk AKYILDIZ**

**Enstitü Anabilim Dalı : Elektronik ve Bilgisayar Eğitimi**

**Bu tez 15/06/2007 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.**

**Yrd. Doç. Dr. Ali Fuat BOZ**  
**Jüri Başkanı**

**Prof. Dr. Abdullah FERİKOĞLU**  
**Üye**

**Doç. Dr. Raşit KÖKER**  
**Üye**

## **TEŐEKKÜR**

Tezin hazırlanış aőamasında bana her türlü desteęi veren danıőman hocam Sayın Yrd. Doę. Dr. Ali Fuat BOZ'a, uygulamanın yapımı aőamasında bana yardımlarını esirgemeyen Mustafa ALP'e ve dięer iő arkadaşlarıma ve ęalıőmalarım esnasında sürekli yanımda olan sevgili eőime sonsuz teőekkürlerimi sunarım.

Mayıs 2007

Faruk AKYILDIZ

## İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER.....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vii
ŞEKİLLER LİSTESİ.....	ix
TABLolar LİSTESİ.....	x
ÖZET.....	xi
SUMMARY.....	xii

### BÖLÜM 1.

GİRİŞ.....	1
------------	---

### BÖLÜM 2.

AKVARYUM HAKKINDA GENEL BİLGİLER .....	4
2.1. Akvaryum Kurulumu ve Dekorasyonu.....	4
2.1.1. Örnek akvaryum ağırlığının hesaplanması.....	6
2.1.2. Güçlendirici kayıtlar.....	6
2.1.3. Dikkat edilmesi gereken hususlar.....	7
2.2. Akvaryumun İç Dekorasyonu.....	8
2.3. Aydınlatma.....	10
2.3.1. Aydınlatma temel bilgileri.....	10
2.3.1.1. Aydınlanma Miktarı (Lux).....	11
2.3.1.2. Işığın sıcaklık derecesi (kelvin).....	12
2.3.1.3. Lambanın renk renderleme faktörü .....	12
2.3.1.4. Lambanın yaydığı ışık dalga boyları (ışık spektrumu).....	12
2.3.2. Estetik aydınlatma.....	13
2.3.3. İhtiyaç aydınlatması.....	13



2.3.3.1. Balık aydınlatması.....	14
2.3.3.2. Bitki aydınlatması.....	14
2.3.4. Lamba tipleri.....	15
2.3.4.1. Elektrik ampülü.....	15
2.3.4.2. Halojen lambalar.....	15
2.3.4.3. Floresan lambalar.....	15
2.3.4.4. Cıva ve sodyum buharlı lambalar.....	17
2.3.4.5. Metal halide.....	17
2.4. Akvaryum Kumu.....	17
2.5. Isıtıcı.....	18
2.6. Hava Motoru.....	19
2.7. Filtreler ve Filtreleme.....	19
2.7.1. Neden filtre?.....	19
2.7.1.1. Biyolojik filtrasyon .....	20
2.7.1.2. Mekanik filtrasyon.....	21
2.7.1.3. Kimyasal Filtrasyon.....	21
2.7.2. Filtre çeşitleri.....	21
2.7.2.1. Sünger filtreler.....	21
2.7.2.2. Kutu filtreler.....	22
2.7.2.3. Kumaltı filtreleri (taban filtresi).....	22
2.7.2.4. Motorlu filtreler.....	23
2.7.3. Nitrat önleyiciler.....	24
2.7.3.1. Kum filtreleri.....	24
2.7.3.2. Bitkiler.....	24
2.7.4. Filtrelemede unutulmaması gerekenler.....	24

### BÖLÜM 3.

PIC 16F877 MİKRODENETLEYİCİ.....	26
3.1. PIC nedir?.....	26
3.2. Neden PIC?.....	27
3.3. PIC16F877 özellikleri ve yapısı.....	29
3.4. PIC16F877'nin besleme uçları ve beslenmesi.....	32
3.5. PIC16F877'nin reset uçları.....	32

3.6. PIC16F877'nin saat uçları ve osilatör tipleri.....	32
3.6.1 Kristal osilatör / Seramik rezonatör.....	33
3.6.2. RC Osilatör.....	33

#### BÖLÜM 4.

AKVARYUM OTOMASYONU TASARIMI.....	35
4.1. PIC 16F877 Mikrodenetleyici.....	42
4.2. 4x16 Likit Kristalli Ekran.....	42
4.3. Tuşlar.....	43
4.3.1. Menü (M).....	44
4.3.2. Aydınlatma (A).....	44
4.3.3. Hava motoru (H) .....	44
4.3.4. Yemlik 1 (Y <sub>1</sub> ).....	44
4.3.5. Yemlik 2 (Y <sub>2</sub> ).....	44
4.4. Giriş Devreleri.....	45
4.4.1. Enerji kontrol.....	45
4.4.2. Su seviye rölesi.....	47
4.5. Çıkış Devreleri.....	49
4.5.1. 220V ile çalışan çıkış devreleri .....	49
4.5.1.1. Hava motoru.....	49
4.5.1.2. Aydınlatma.....	50
4.5.1.3. Isıtıcı.....	50
4.5.1.4. Dip süpürme motoru.....	50
4.5.1.5. Kirli su valfi.....	51
4.5.1.6. Temiz su valfi.....	52
4.5.2. 5V ile çalışan çıkış devreleri.....	52
4.6. Sistemdeki Diğer Devreler ve Elemanlar.....	54
4.6.1. DS1302 saat ve takvim devresi.....	54
4.6.2. Uzaktan kumanda ve IR (Uzaktan kumanda alıcısı – TK19).....	55

#### BÖLÜM 5.

SONUÇLAR VE ÖNERİLER.....	56
---------------------------	----

KAYNAKLAR.....	58
EKLER.....	59
ÖZGEÇMİŞ.....	109

## SİMGELER VE KISALTMALAR LİSTESİ

PIC	: Peripheral Interface Controller - Çevresel arabirim denetleyicisi
LCD	: Liquid Crystall Display – Likit kristal ekran
UPS	: Uninterruptible Power Supply - Kesintisiz güç kaynağı
nm	: Nanometre
mm	: Milimetre
cm	: Santimetre
~	: Yaklaşık
kg	: Kilogram
%	: Yüzde
pH	: Hidrojen konsantrasyonunun eksi logaritması
W	: Watt
K	: Kelvin
CO <sub>2</sub>	: Karbondioksit
O <sub>2</sub>	: Oksijen
&	: Ve
°C	: Santigrat derece
ΔT	: Sıcaklık farkı
T <sub>1</sub>	: Odanın sıcaklığı
T <sub>2</sub>	: Akvaryumdaki su sıcaklığı
>	: Büyük
R	: Direnç
C	: Kondansatör
P	: Potansiyometre
Q	: Transistör
D	: Diyot
T	: Triyak
V	: Volt

$\Omega$	: Ohm
K $\Omega$	: Kilo ohm
Hz	: Hertz
KHz	: Kilo Hertz
MHz	: Mega Hertz
pF	: Piko Farad
$\mu$ F	: Mikro Farad
nF	: Nano Farad
H	: Hava motoru
M	: Menü
A	: Aydınlatma
Y <sub>1</sub>	: 1. Yemlik
Y <sub>2</sub>	: 2. Yemlik
mA	: mili amper
CMOS	: Metal Oxide Semiconductor-Bütünleyici metal oksit yarıiletken
RAM	: Random access memory - Rasgele erişimli bellek
ROM	: Read Only Memory - Sadece okunur bellek
EPROM	: Erasable Programmable Read Only Memory - Silinir, yazılır sadece okunur bellek
EEPROM	: Electrically Erasable Programmable Read Only Memory - Elektrikle Silinir, yazılır sadece okunur bellek
Ü	: Üst
A	: Alt
T	: Toprak
L	: Faz
N	: Nötr
IR	: Infrared – Kıızıl ötesi

## ŞEKİLLER LİSTESİ

Şekil 1.1. Akvaryum otomasyonunun genel yapısı.....	3
Şekil 2.1. Dış filtre.....	23
Şekil 3.1. PIC16F877 Bacak tanımlamaları.....	30
Şekil 4.1. Sistemin blok şeması.....	35
Şekil 4.2. Anakart devre kutusunun önden görünüşü.....	36
Şekil 4.3. Anakart devre kutusunun arkadan görünüşü.....	36
Şekil 4.4. Göstergeli kontrol paneli.....	37
Şekil 4.5. PIC 16F877 mikrodenetleyicili anakart devresi açık şeması.....	38
Şekil 4.6. Göstergeli kontrol paneli devresi açık şeması.....	39
Şekil.4.7. Güç kaynağı.....	45
Şekil 4.8. Kesintisiz güç kaynağı (UPS).....	46
Şekil 4.9. Enerji kontrol devresi.....	46
Şekil 4.10. Su seviye rölesi.....	47
Şekil 4.11. Su seviye rölesinin devre şeması.....	48
Şekil 4.12. Su boşaltma motoru.....	48
Şekil 4.13. 220V ile çalışan çıkış devrelerin açık şemaları.....	49
Şekil 4.14. Hava motoru.....	49
Şekil 4.15. Flüoresan.....	50
Şekil 4.16. Isıtıcı.....	50
Şekil 4.17. Dip süpürme motoru.....	51
Şekil 4.18. Kirli su valfi.....	51
Şekil 4.19. Temiz su valfi.....	52
Şekil 4.20. 5V ile çalışan çıkış devrelerin açık şemaları.....	53
Şekil 4.21. Yemlik.....	53
Şekil 4.22. DS1302 Tüm devresi.....	54
Şekil 4.23. Uzaktan kumanda (a), Uzaktan kumandanın tuş takımı (b).....	55
Şekil 4.24. TK19.....	55

## TABLolar LİSTESİ

Tablo 2.1. Akvaryum cam kalınlıkları .....	7
Tablo 2.2. Akvaryum hacmine göre ısıtıcı güçleri.....	19
Tablo 3.1. Neden PIC?.....	27
Tablo 3.2. PIC16F877 ve PIC16F84 işlemcilerinin kıyaslanması.....	29
Tablo 3.3. Osilatör çeşitleri.....	33
Tablo 3.4. Frekansa göre kondansatör seçimi.....	33
Tablo 4.1. Anakart devresinin eleman listesi .....	40
Tablo 4.2. Göstergeli kontrol paneli devresinin eleman listesi.....	41

## ÖZET

Anahtar Kelimeler: Akvaryum Otomasyonu, Mikrodenetleyici, PIC 16F877

Bitkilerle, kayalarla ve ağaç kökleriyle dekore edilmiş güzel bir akvaryumda yüzen renkli balıkları izlemek eğlenceli bir iştir. Akvaryum, yaşadığınız evde, iş yerinde, vaktinizin çoğunu geçirdiğiniz kapalı bir alanda, sessizce devam eden, apayrı bir dünyadır. Akvaryum dinlendirir, stresi azaltır. Hatta akvaryuma bakanların kalp atışlarının yavaşladığı bilimsel bir gerçektir.

Bir akvaryumun yemleme, filtreleme, aydınlatma, havalandırma, ısıtma ve su değişimi ihtiyaçları vardır. Balıkların zamanında ve gerektiği kadar yemlenmesi gereklidir. Yine akvaryum suyunun zamanında ve gerekli miktarda su değişiminin yapılması gereklidir. Balıkların bu bakımlarının düzenli yapılabilmesi ve yoğun iş yükü olan veya uzun süreli gezilere çıkan insanların evlerinde bulunan akvaryumlarının bakımı için bir akvaryum otomasyonuna ihtiyaç duyulmuştur. Yapılan akvaryum otomasyonunun elektronik kısmı iki bölümden oluşmaktadır. Birincisi PIC 16F877 mikrodenetleyicili anakart devresi, ikincisi göstergeli kontrol panelidir. Kullanıcı işlemlerini göstergeli kontrol panelindeki 4x16 LCD (Liquid Crystal Display) ve 5 adet tuş yardımıyla kolaylıkla yapabilmektedir. Bu tuşlar, yemlik 1, yemlik 2, aydınlatma, hava motoru ve menüdür. Tasarımda iki adet yemlik vardır. Bu yemliklerden bir tanesi yavru balıklar için tasarlanmıştır. Yemleme; uzaktan kumandadan, göstergeli kontrol panelinden ve önceden programlanan zamanda yapılabilmektedir. Aydınlatma ve hava motoru, hem uzaktan kumandadan hem de göstergeli kontrol panelinden kontrol edilebilmektedir. Su değişimi haftada bir kez yapılmaktadır. Bunun zamanı önceden göstergeli kontrol panelinden ayarlanmaktadır.

PIC 16F877 mikrodenetleyicili anakart devresi ve göstergeli kontrol paneli devresinin çizimi ve simülasyonu Proteus ISIS Professional 6.9 SP5 programı ile yapılmış; baskı devreleri ise Proteus ARES Professional 6.9 SP5 programı ile hazırlanmıştır. Tasarımın yazılımı MPLAB IDE 7.50 programı ile yapılmış; yazılımın PIC 16F877 mikrodenetleyicisine aktarılmasında EPICWIN programı kullanılmıştır. Yazılım HI-TECH PICC Toolsuite dili kullanılarak hazırlanmıştır.



# **DESIGN AND PRODUCTION OF AN AQUARIUM AUTOMATION SYSTEM**

## **SUMMARY**

Key words: Aquarium automation, Microcontroller, PIC16F877,

It is very exciting to watch coloured swimming fish in a beautiful aquarium that is organized with plants, rocks and tree roots. Aquarium is a different world that goes on silently in your home, covered area that you pass the time. Aquarium allows to rest, reduces stress. Also it is a reality aquarium slows down heartbeats.

According to investigation an aquarium needs feeding, filtering, illumination, ventilation, heating and water changing. It is necessary that fish must be fed on time and as required. Aquarium's water changing must do on time. Because of this aquarium automation feels to need for people that has a job and go to holiday. Aquarium automation's electronic section occurs two parts. First PIC 16F877 mainboard circuit with microcontroller second, control panel with indicator. Indicators that are on user processes can be made with 4x16 LCD and 5 number keys. These keys are manger 1, manger 2, illumination, air motor and menu. There are two mangers on design. One of them was planned for young fish. Feeding can be made on time with ir remote, control panel with indicator. Illumination and air motor can be controlled with ir remote and control panel with indicator. Water changing is made once a week. It is set control panel with indicator.

PIC 16F877, mainboard circuit with microcontroller and control panel with indicator circuit's drawing and simulation was made with Proteus ISIS Professional 6.9 SP5. PCB was designed with Proteus ARES Professional 6.9 SP5. Software was written with MPLAB IDE 7.50 and programme was transferred with EPICWIN software. Software was prepared with HI-TECH PICC Toolsuite language.

In first section aims of this work and other works are explained. In Second section, some information about aquarium was given. In third section electronic circuit component that are used in design was explained. In last section conclusion and suggestions were explained.

## BÖLÜM 1. GİRİŞ

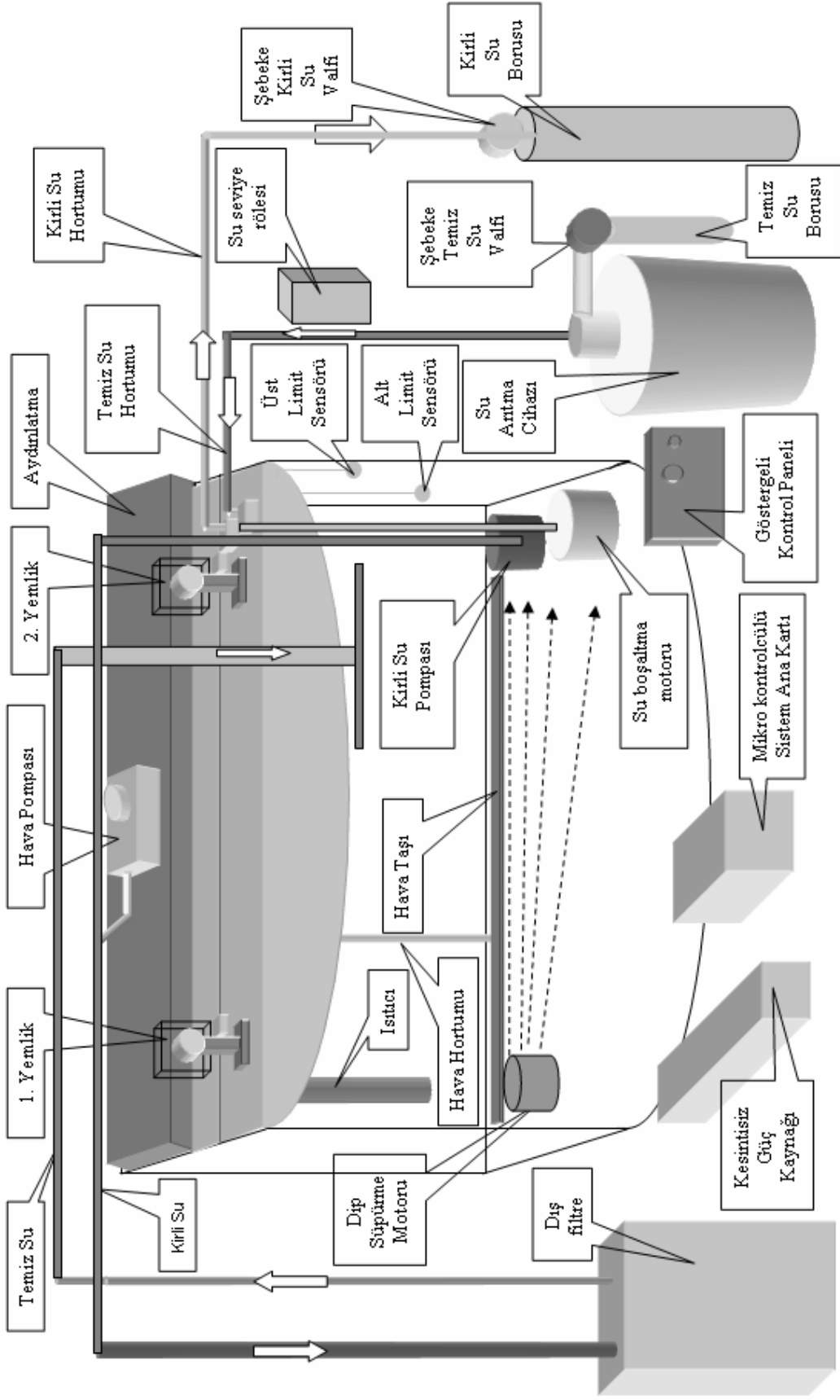
Akvaryum, yaşadığınız evde, iş yerinde, vaktinizin çoğunu geçirdiğiniz kapalı bir alanda, sessizce devam eden, kendi telaşesiyle varlığını sürdüren, apayrı bir dünyadır. Bitkilerle, kayalarla ve ağaç kökleriyle dekore edilmiş güzel bir akvaryumda yüzen renkli balıkları izlemek ayrı bir zevktir. Akvaryum dinlendirir, stresi azaltır. Hatta akvaryuma bakanların kalp atışlarının yavaşladığı bilimsel bir gerçektir[1]. Bir akvaryumda bulunan balıkların aydınlatmaya, havalandırmaya, yemlemeye, filtrelemeye, su değişimine ve ısıtıcıya ihtiyaçları vardır. Balıkların zamanında ve gerektiği kadar yemlenmesi gereklidir. Yine akvaryum suyunun zamanında ve gerekli miktarda su değişiminin yapılması gereklidir. Balıkların bu bakımlarının düzenli yapılabilmesi ve yoğun iş yükü olan veya uzun süreli gezilere çıkan insanların evlerinde bulunan akvaryumlarının bakımı için bir akvaryum otomasyonuna ihtiyaç duyulmuştur. Daha önce yapılan çalışmaların varlığı araştırıldığında kaynak [2]'de belirtilen bir akvaryum otomasyonu sistemi ile karşılaşılmıştır. İlgili sistemde akvaryumun filtre, hava motoru ve aydınlatma birimlerini kontrol edilmektedir. Kullanıcı bu birimlerin açık ve kapalı kalma sürelerini belirlemektedir.

Bu tezin amacı, yoğun iş yükü olan veya uzun süreli gezilere çıkan insanların evlerinde bulunan akvaryumlarının, önceden hazırlanmış programa göre çalışmasını sağlamaktır. Bu çalışmada akvaryumda bulunan balıkların aydınlatma, havalandırma, yeme, filtreleme, su değişimi ve ısınma ihtiyaçları akvaryum otomasyon sistemi ile karşılanmaktadır.

Akvaryum otomasyonu tasarımının genel yapısı Şekil 1.1'de verilmiştir. Akvaryum otomasyonunun elektronik devresi, PIC 16F877 mikrodenetleyicili anakart ve göstergeli kontrol paneli devrelerinden oluşmaktadır. Kullanıcı işlemlerini göstergeli kontrol panelindeki yemlik 1, yemlik 2, aydınlatma, hava motoru ve menü tuşlarıyla

ve 4x16 LCD (Liquid Crystal Display) yardımıyla kolaylıkla yapabilmektedir. Tasarıma eklenen UPS (Kesintisiz Güç Kaynağı) sayesinde yemleme ve havalandırma işlemleri elektrik kesildiğinde devam etmektedir.

Akvaryum hakkında binmesi gereken genel bilgiler bölüm 2’te açıklanacaktır. PIC 16F877 mikrodenetleyicisi hakkında geniş bilgi bölüm 3’te verilecektir. 4x16 LCD (Liquid Crystal Display), tuşlar, yemlik 1, yemlik 2, hava motoru, aydınlatma, ısıtıcı, kirli su valfi, temiz su valfi, su seviye rölesi, dip süpürme motoru, kesintisiz güç kaynağı, uzaktan kumanda, IR (TK 19), LM 35 sıcaklık algılayıcısı, buzzer, DS 1302, giriş ve çıkış devrelerin çalışması ve sistemin çalışması hakkında geniş bilgi bölüm 4’te verilecektir. Dördüncü bölüm sonuç ve önerilerden oluşmaktadır.



Şekil 1. Akvaryum otomasyonunun genel yapısı

## **BÖLÜM 2. AKVARYUM HAKKINDA GENEL BİLGİLER**

### **2.1. Akvaryum Kurulumu ve Dekorasyonu**

Bir akvaryum kurulumunda ilk yapılacak işlem, akvaryumun evin hangi köşesine konulacağına karar vermektir. Bu bölge kesinlikle direk güneş ışığı almamalı, akvaryumla uğraşırken hareket kabiliyetini kısıtlamamalı, aynı zamanda çok ayakaltında olmamalıdır. Ayrıca akvaryum bulunan odanın nem oranının yüksek olacağı unutulmamalıdır. Akvaryumumuzun sehpa dolaplı olması, bize ilerde her şeyin elimizin altında olmasını sağlayacaktır. Dolap yüksekliğinin oturulan yerden rahatça seyredilebilecek hizada olması görünümü güzelleştirecektir. Unutulmaması gereken bir nokta da sehpa ile akvaryum arasına ince bir köpük (strafor) koymaktır, bu zemindeki ve sehpadaki titreşimleri emerek balıkların paniklemesini önler. Diğer bir faydası da akvaryumun ağırlığıyla köpük dengeli olarak sıkışıp, yapım sırasında bir ölçsüzlük varsa bunu minimuma indirmesidir.

Akvaryumun alabileceği en büyük olanı seçildiğinde, bakımı rahat olmaktadır. Boyutlar büyüdükçe akvaryumun bakımı kolaylaşır. Sudaki kimyasal dengeler kolay kolay bozulmaz. Dekorasyon için de zorlanılmaz. Arka cama koyu renkli karton veya dekoratif kaplama koymakta fayda vardır. Bu balıkların boşlukta kalma hissini ortadan kaldırır, elektrik kablolarının ve hava borularının gözükmesini önler. İleride arka cam yosun yaparsa doğal bir fon oluşturacaktır.

Akvaryum ve malzemelerinin temizliğinde kesinlikle deterjan ve türevleri kullanılmaz. Temizlik için çeşme suyu, fırça (diş fırçası), jilet (gerekirse), kaya tuzu (iyotsuz sofr tuzu da olabilir) kullanılır. Yeni akvaryum bol miktarda çeşme suyu ile iyice yıkanır, temiz bir bez ya da kağıt havlu ile kurulur, sonra yere bir kuru kağıt (gazete) serilir, akvaryum kağıdın üzerine konur ve yavaşça doldurulur. Bu sızdırmazlık testidir eğer akvaryum su sızdırıyorsa sızan su kağıt üzerinde abartılı

bir şekilde görülecektir.

Her şey hazırlanınca akvaryum yerine konur, öncelikle hava hortumları vantuzlarla tutturulur, büyük dekoratif malzeme varsa alta veya arka cama akvaryum için üretilen silikonla tutturulur. Filtrenin yeri tespit edilir, dış filtre ise emiş ve basış hortumlarının yerleri belirlenir. Isıtıcının yeri her zaman filtrenin basış yerine yakın olmalıdır ki ısı akvaryum içinde düzgün olarak dağılabilsin. Termometre ise her zaman ısıtıcıya en uzak mesafede olmalıdır. Kum veya çakıl yavaşça tabana yayılır kalınlık genelde 4 cm – 5 cm arasında yapılır ama bu kalınlık akvaryumun büyük veya küçüklüğü ile azalır çoğalır. Arka taraf yüksek ön taraf alçak yapılırsa önden görünüşte akvaryuma bir derinlik hissi verilebilir.

Akvaryuma su konulurken, suyun yapılan taban çalışmalarını bozmaması için kumun üzerine bir tabak tabağın içine de bir bardak konulursa su bardağa ne kadar hızlı dökülürse dökülsün tabana bir zarar vermez. Su seviyesi yarıyı geçtikten sonra bitkiler yerlerine yerleştirilir. Bu arada boyu uzayan ve çabuk büyüyen bitkiler arka plana boyu kısa ve yavaş büyüyen bitkiler ön tarafa konulur.

Akvaryum tamamen dolunca filtre, hava motoru, ısıtıcı çalıştırılır. İlk başta berrak olan su bir iki gün içinde bulanabilir bu önemli değildir. Su içindeki biyolojik denge yavaş yavaş kuruldukça su berraklaşmaya başlar. Balıklar için acele etmemek lazımdır, eğer hassas su dengesi isteyen balık koyacaksak 15 – 30 gün sabretmek balıkların sağlığı için yararlı olacaktır [3] .

Akvaryum duvara yapıştırılmamalıdır. Her zaman ~10cm aralık bırakılmalıdır. Şelale (askı) tipi filtre kullanılacaksa bu mesafe daha da uzun tutulmalıdır. Bu hem duvara daha az su sıçraması hem de daha sonra akvaryumun arkasına zor da olsa ulaşabilmeyi sağlar [4] .

Her şeyden evvel, cam kesimi ve akvaryum yapımı, profesyonellere bırakılması gereken bir iştir, ustalık ve tecrübe ister. Ufak ölçüm ve yapıştırma hataları felaketlere yol açar, bu yüzden az bir ekonomi sağlamak için alınan riske kesinlikle değmez. Bütün sistemleriyle birlikte hesaplandığında, bir akvaryum sisteminin

maliyetinde akvaryumun kendisinin maliyeti o kadar yüksek tutmayacağı için, burada sağlanacak maliyet düşüşü de tüm sistem hesaplandığında yüksek olmayacaktır.

Akvaryum ( ve içindeki su, kayalar, vs. ) ciddi ağırlıktadır, bu yüzden akvaryumu taşıyacak mobilyanın da bu cins bir ağırlığı rahatça taşıyabilecek nitelikte olması gerekir.

### **2.1.1. Örnek akvaryum ağırlığının hesaplanması**

Akvaryum sisteminin ağırlığı: Akvaryum camı, su, kum, çakıl, kayalar, akvaryum kabineti ve filtrasyon sisteminin ağırlığı'nın toplamıdır.

Akvaryum sisteminin tabana uyguladığı basınç: Akvaryum sisteminin ağırlığı / kabinet ayaklarının taban alanı 'dır.

Ağırlığı 500 kg ' ı aşan akvaryum sistemlerinin odanın köşelerinde kurulu olması önerilir, çünkü tabanın en sağlam olduğu bölgeler köşelerdir.

### **2.1.2. Güçlendirici kayıtlar**

Akvaryumun ön ve arka camında, boydan boya, akvaryum büyüklüğüne bağlı olarak 3 – 5 cm kalınlığında kayıtlar kesinlikle gereklidir. Enine konulacak kayıtlar da faydalıdır, ancak üst kayıtlar birbirlerine değmemelidir.

Her 60 cm genişlikte bir akvaryumun arka camı ile ön camını üstten birleştiren ara bağlantılar gereklidir. Bağlantının genişliği en az 8 cm civarında olmalıdır, çok geniş olurlarsa, akvaryumda çalışmalarınız sırasında engelleyici olurlar, çok dar olurlarsa yeterince kuvvetli olmazlar.

Akvaryum camının kenarını rodajlamak, hem estetik olarak daha güzel bir görünüm sağlar, hem de camın kenarında kesim sırasında oluşan mikroskobik çatlakları ortadan kaldırarak, bunların ileride büyüyerek camın kırılmasını önler.

Akvaryum cam kalınlıkları Tablo 2.1’de verilmiştir. Burada genişlik ve yükseklik cm, cam kalınlıkları da mm cinsinden verilmiştir. Buradaki değerler biraz güvenliği artırılmış değerlerdir, ama ülkemiz şartlarında gereklidir ( deprem tehlikesi ). Burada değerler riskleri mümkün olabildiği kadar azaltmak için bir parça normalden yükseltilmiştir.

Tabanda kullanılacak cam kalınlığının, yan yüzeylerde kullanılan cam kalınlığından bir adım daha kalın olması önerilir ( mesela yan yüzeyler 12 mm, taban 15 mm veya yan yüzeyler 8 mm, taban 10 mm ).

Tablo 2.1. Akvaryum cam kalınlıkları

		GENİŞLİK																
		20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180
DERİNLİK	90	12	12	12	12	12	12	12	12	12	15	15	15	15	15	15	15	15
	80	10	10	10	10	10	12	12	12	12	12	12	12	15	15	15	15	15
	70	10	10	10	10	10	10	10	12	12	12	12	12	12	12	12	12	12
	60	6	8	10	10	10	10	10	10	10	12	12	12	12	12	12	12	12
	50	6	6	8	8	10	10	10	10	10	10	10	10	12	12	12	12	12
	40	5	6	6	6	8	8	10	10	10	10	10	10	10	10	10	12	12
	30	5	5	5	6	6	6	8	8	8	8	10	10	10	10	10	10	10
	20	4	5	5	5	5	5	6	6	6	6	6	8	8	8	8	8	8

### 2.1.3. Dikkat edilmesi gereken hususlar

Üstü açık akvaryumun avantajları:

- Daha yüksek güçlü ışıklandırma kullanılabilir.
- Zehirli veya faydalı daha çok gaz alışverişi vardır.
- Bitkiler büyüyerek akvaryumun dışına taşabilir.
- Bitkiler ve balıklar üstten bakarak ta seyredilebilir.
- Bakımı ve kontrolü daha kolaydır, özellikle akvaryum yüksek ise.
- Işıklar akvaryumdan daha uzağa monte edilebileceği için, yaydıkları ısının suyu ısıtması büyük ölçüde önlenir.

Üstü açık akvaryumların dezavantajları:

- Balıklar dışarıya zıplayabilir.(Fakat bu %90 oranında akvaryum üstünde 5 cm



kalınlığında kullanılacak kuşaklarla önlenebilir. Balıklar genelde akvaryumu boydan boya yüzdükten sonra cama çarpınca yukarı doğru hareketlenip sıçrarlar, eğer akvaryumun üst kenarına içten cam kuşak kullanılmış ise, balıklar bu kuşağa çarpıp suya geri düşeceklerdir. )

- b) Yüksek su buharlaşma oranının getirdiği aşırı nem tehlikesi.
- c) Toz akvaryumun içine daha çok girer.
- d) Genelde bu tip akvaryumların üzerlerine asılan ışıklar, akvaryumla birlikte odayı da aydınlatırla, özellikle yüksek güçlü metal halide lambalar.

Akvaryum yapımında, içindeki zehirli maddeler çok azaltılmış cins silikon kullanılması gerekir. Piyasada akvaryum yapımı için üretilmiş, çok az zehirli madde içeren silikonlar mevcuttur. Akvaryum camını silikon kullanarak yapıştırmak, yine profesyonellere bırakılması gereken bir iştir. Camlar arasındaki silikon tabakanın kalınlığı 1 milimetre civarında olmalıdır, ne daha az, daha da önemlisi ne daha fazla. Silikonun 24 saat kuruma süresi vardır ve 3 kullanımdan 3 gün sonra tamamen güvenlidir. Satın alınan silikonun özelliklerini anlatan kullanma talimatlarını okumak burada fayda sağlar, değişik marka silikonların değişik kuruma ve kullanım özellikleri olabilir. Yapıştırma ve kuruma bittikten sonra, akvaryumu 1 hafta süreyle, çeşme suyuyla tam dolu olarak, içinde su pompaları suyu iyi sirküle edecek şekilde çalışarak test edilmesi faydalı olacaktır, bu hem sızıntıları kontrol etmek hem de az da olsa silikonda kalan toksit maddeleri suda ayırıştırmak için çok yararlıdır [5] .

## 2.2. Akvaryumun İç Dekorasyonu

Dekorasyon için her türlü malzeme kullanılabilir, yalnız bu malzemelerin suda erimemesi, suyla reaksiyona girmemesi, kireçli bir yapıda olmaması, toksin olmaması, suda çözünen ağır metalleri ihtiva etmemesi gerekir. Piyasada seramik bitki kökleri, kayalar, çeşitli yuva kovukları ve doğal bitki kökleri, taşlar, kayalar mevcuttur [3] .

Akvaryuma konulacak malzemeler, akıntıyı engellememelidir. Özel durumlar hariç, tankın hiçbir yerinde akıntısız alan olmamalıdır. Aksi halde tüm atıklar burada birikir ve buradakiler gaz değişiminden yoksun kalır.

Bazı canlı türlerinde belli çiftler veya grupların birbirini daha az görmesi uygun görülür. Özellikle cichlidler ve deniz canlılarında bu durum söz konusudur. Bunun için genelde serbest alan içinde taş, kütük v.s. ile bölmeler kurulur. İşte bu bölmeler arasında kalan alanlara ayrı ayrı akıntı yetiştirebilecek düzeyde bir filtre çıkışı veya ek akıntı pompaları düşünülmelidir.

Akvaryumun taban malzemesi önden arkaya hafif eğim yaratacak şekilde dökülür. Bu hem görsel estetik açısından ve arkaların görülmesi için yapılıp hem de atıkların önlere doğru gelmesini kolaylaştırmak içindir.

Boy sırasına göre dizilim özellikle bitkilerde çok önemlidir. Mesela *Vallisneria spiralis* (Saz) veya *Egeria densa* (Elodea) gibi hızlı uzayan bir bitki ön bölgeye ekilmemelidir. Aksi halde 1–2 ay içinde ön camı yeşil bir duvar kaplar. *Cryptocoryne* türleri gibi genelde kısa boylu bitkiler de arkalarda ekilmemelidir. Yoksa önlerinde yükselecek bitkiler yüzünden hiç görülmezler. Bol ışık gerektiren bitkiler hızlı uzayan veya geniş yaprakları olanların yakınında ekilmemelidir. Az ışık gerektirenlerde ise aksi yapılmalı yani diğerlerinin gölgesinde ekilmelidir.

Taşlarda iki özellik önemlidir. Birincisi şekilleri akıntıya ve akvaryumdaki canlıların yaşam tarzına uygun olmalıdır. Bazı cichlidler için düz taşlar, bazıları için ise bol sayıda delik ve mağara yaratacak çıkıntılı olanlar seçilmelidir. Deniz akvaryumunda taşlar olabildiğince boşluk ve deliklerle dolu olmalı ve çok sayıda mağara oluşturulmalıdır. Taşların rengi de önemlidir. Canlılar genelde koyu renkli desenleri tercih ederler. İkincisi taşların kimyasal bileşiği çok çok önemlidir. Çok masum ve temiz görünen bir kayrak taşının çatlağında bol miktarda kükürt bileşiği birikmiş olabilir. Görünen çatlaklar kontrol edilmeli ve temizlemeye çalışılmalıdır. Taşların sertliğini kontrol edilmelidir. Suya girince çatlayacak bir taşın içinden neyin çıkacağı bilenemez. Tüm bu önlemlere rağmen taşlar, özellikle büyük olanlar akvaryuma konmadan önce suda uzun süre bekletilmelidir. Mümkünse sıcak suyla yıkanmalıdır.

Kireç içeren taşlar ve yumuşak yapıları olanlar suyu sertleştirecektir ve pH'ı da yükseltebilir.

Kütükler konusunda iş daha kolaydır. Piyasada 2 tür kütük var. Biri koyu renkli ve bazen üzerinde çamur olan kütüktür. Genelde çamurun altında kıvı-kahve renkli ve şık bir ağaç deseni bulunur. Akvaryum için tercih edilecek kütük budur. Bunlar bataklık ve çamurlu alanlarda yıllarca çürümüş yaşlı ağaç köklerinden çıkarılan parçalardır. Genelde tropikal sık ormanlardan çıkartılır. Testere ile bile kesilmeyecek kadar sert yapısı vardır. İşte bu yüzden bunca yıl çürümeye dayanabilir. Diğer kütük türü ise daha açık renkli, soyulmuş kuru ağaç dalı görünümünde olandır. Orijinal olana benzetmek için bunu da çamura gömebilirler veya çürüyen sıradan ağaç köklerinden alabilirler ama akvaryum için uygun değiller. Temiz olanları belki kısa süre için kullanılabilir ama uzun süre kalmamalı çünkü çürüyecektir.

Akvaryumda kütük kullanmak suyun pH'ını az ya da çok indirecektir. O yüzden her akvaryum türünde kütük kullanılmamalıdır.

Ayrıca akvaryum suyu ile temasta olan her materyalin iyi tanınması gereklidir. Materyaller, suya herhangi bir şey bırakıp bırakmadığını öğrenmeden kullanılmamalıdır. Örneğin plastik malzemenin bile zararlısı var. Bazı adi plastik malzemeler suya zehirli bileşikler bırakabilir. Bağlantı malzemelerinizi temiz su tesisatında kullanılan ürünlerden seçilmeli veya akvaryuma özel üretilen (genelde filtre üreten firmalar) ürünler kullanılmalıdır [4] .

## **2.3. Aydınlatma**

Akvaryumlarda aydınlatma hem estetik açıdan hem de bitkiler için artı, canlıların kendilerini daha iyi hissetmeleri (doğal) için gereklidir. Bu yüzden aydınlatma estetik ve ihtiyaç açısından ele alınabilir.

### **2.3.1. Aydınlatma temel bilgileri**

Aydınlatmayı anlayabilmek için aydınlanma miktarı (lux), ışığın sıcaklık derecesi (kelvin), lambanın renk renderleme faktörü ve lambanın yaydığı ışık dalga boyları (ışık spektrumu) temel bilgilerin anlaşılması gerekmektedir.

### 2.3.1.1. Aydınlanma Miktarı (Lux)

Bir ışık kaynağından (bizim durumumuzda lamba) belli bir mesafedeki ışık şiddetidir. Aydınlanma miktarının birimi lux'dür. Lambanın ışık yoğunluğu birçok lambanın üzerinde bulunmaktadır. Işık yoğunluğunun birimi lümen'dir. (Mesela Kompakt 20Watt Eko bir floresan üzerinde 1200 lümen yazmaktadır.) Bu konunun açıklanma sebebi birçok yerde yapılan hatayı düzeltmek içindir. Akvaryum için gerekli aydınlatma sorulduğunda genellikle litre başına şu kadar watt koy gibi ifadeler veya tavsiyelerde bulunulur. Watt bir güç birimidir. Işık şiddeti veya miktarı ile hiç bir ilişkisi yoktur. Şöyle ki; normal ampul tükettiği enerjinin (mesela 100 Watt) sadece % 6'sını (evet sadece %6'sı) ışık üretmek için kullanmakta geri kalanı ile ısı üretilmektedir. Metal Halide lambalarda ise bu değer %30–40 arasındadır. Yani 100 Wattlık bir metal halide ampul yaklaşık 600–700 Wattlık ampulün çıkardığı ışığı üretmektedir. Şöyle izah etmek gerekir ise 100 Wattlık bir ampulde görünür ışık sadece 5-6Watt'tır. Geri kalanı ise kızıl ötesi ışınlar ve ısıdır.

36 Wattlık bir TL'D serisi Philips floresanda ise toplam görünür ışık yayılması 10Watt geri kalanı ise infrared (kızıl ötesi ışınlar) ve diğer kayıplardır.

Philips'in yine 400 Wattlık bir Metal Halide lambasında ise görünür ışık yayılımı 96 Watt, infrared yayılımı 237 Watt diğer kısmı ise ısı üretimidir. Görünür ışık yayılımı Metal Halide lambalar için güç arttıkça artmaktadır. (2000 Wattlık Metal Halide için 800–900 Watt civarı. Geri kalanı infrared ve ısı. Bu yüzden küçük güçlü metal halide lambası pek tavsiye edilmemektedir.)

Üstelik burada daha önemli bir konu var. Akvaryuma gereken ışık, tabanın genişliği ile doğru orantılı, akvaryum derinliğinin ise karesi ile orantılı olmalıdır. Özellikle lambadan uzaklaştıkça aydınlanma miktarı mesafenin karesi ile azalır. Yani lambayı ne kadar uzağa koyarsanız veya akvaryum ne kadar derin ise aydınlanma miktarı o kadar azalır. Bunun yanında akvaryum suyu özellikle ultraviyole ve bazı bitkiler için faydalı ışınları süzmektedir. (Güneş gözlüğü gibi davranmaktadır.) Bu yüzden derinlik arttıkça lamba adeti veya gücü artırılmalıdır. Bu anlatılanlardan anlaşılacağı gibi akvaryumda ışık seçimi watt/litre hesabı değil de tecrübe ve bilgi gerektirmektedir.

### 2.3.1.2. Işığın sıcaklık derecesi (kelvin)

Işığın rengi kelvin cinsinden ifade edilir. Işığın rengi şu demektir: Bir tav fırınında veya bir Demir-Çelik Fabrikasında potadan dökülen demirin rengini düşünün. Bu potada dökülen demirin sıcaklığı, yaydığı ışığın da rengini ifade eder. Bu konunun akvaryumda nasıl kullanılacağına gelince: Gün ışığının sıcaklık derecesi 5000 Kelvin civarındadır. Pratik olarak kullanılan aydınlatma apareyleri de (lamba, floresan vs.) bu sıcaklık derecesi civarı veya daha yüksek olmalıdır. Güneş ışığı gün boyunca değişik sıcaklıklarla yere geliyor. Bunu sebebi, atmosfere vuruş açısının değişmesidir. Sabahları soğuk (düşük kelvin) ışıktan başlayarak, öğlen ve öğleden sonra özellikle tropik bölgelerde 10,000K'i aşar. Batımına doğru da yine sıcaklıkta düşüş görülür. Otomatik zamanlayıcı sistemler ve değişik derecelerdeki ampullerle doğayı taklit etmek mümkündür. Bu durumda aydınlatma için ayrılan bütçenin artması demektir. Buna pek fazla gerek yoktur. Gün boyu gelen ışığın ortalama sıcaklığını oluşturacak sürekli bir sistemden hemen hemen aynı verimi alınır. İşte bunun için genel aydınlatmanız ortalama 5500–7500 arası olmalıdır.

### 2.3.1.3. Lambanın renk renderleme faktörü

Renkler değişik lamba tiplerinde farklı görülürler. Bu konuyu kolayca anlatmak istersek: floresan lambalar altında renkler olduğundan farklı görünürler. Normal lambalar ve halojen de ise renkler normal güneş ışığındakinin yaklaşık olarak aynı olarak görülürler. Bu faktör halojende ve normal lambalarda çok yüksek (gün ışığına çok yakın yani 100 e yakın) floresan, metal halide vs.de ise lamba tipine göre 65 ve çok daha aşağıdadır. Bu özetle halojen ve normal ampullerde renkler gerçeğine daha yakın görülürler, diğer tip lambalarda ise renkler olduğundan farklı görülürler demektir.

### 2.3.1.4. Lambanın yaydığı ışık dalga boyları (ışık spektrumu)

Gün ışığı içinde bütün renkleri içerir. Beyaz görünen gün ışığı içerisinde mor, mavi, gök mavisi, yeşil, sarı, turuncu ve kırmızı renkleri bulunmaktadır. Her rengin ayrı dalga boyu vardır. Mavi Işık 400 Nano metre(nm), Sarı 550-600nm, Kırmızı 700-

750nm gibi. Bir ışık kaynağının yaydığı ışıkta içinde çeşitli renkler içerir. İnsan gözünün en hassas olduğu renk sarıdır. Bu renk altında cisimler göz yorulmadan incelenebilir. (Normal ampuller ve halojen lambaların yaydığı ışığın rengine dikkat ediniz) Bu yüzden evlerde kullanılan lambalar bu dalga boylarını en fazla şekilde yansıtacak şekilde tasarlanırlar. Fakat bitkiler için bu dalga boyları neredeyse bir şey ifade etmez. Yukarıda ifade edildiği gibi lambaların üzerinde yaydığı ışık şiddeti yazılır. Fakat bu değer sarı ışık cinsinden ifade edildiği için bizim için çok fazla bir şey ifade etmez. Önemli olan lambanın hangi renklerde ışık yaydığıdır. 450 nm ve 650 nm dalga boyları bitkinin fotosentezini özendirir ve arttıran dalga boyları olduğu için GroLux bu dalga boylarında diğer dalga boylarında olduğunun üç –beş katı daha fazla ışık vermektedir.

### **2.3.2. Estetik aydınlatma**

Estetik açıdan aydınlatma akvaryumdaki bitkilerin ve balıkların renklerinin vurgulanması, akvaryumdaki detayların gözü yormadan izlenmesi amacıyla kullanılır. İhtiyaç aydınlatması ile birlikte kullanılabilir. Estetik aydınlatma için halojen aydınlatma veya gün ışığı floresan lambalar kullanılabilir.

### **2.3.3. İhtiyaç aydınlatması**

Bitkilerin büyümesi ve balıkların doğal ortamlarında bulduklarının hissiyatının verilmesi için aydınlatma gereklidir. Aydınlatma tipi seçilirken bazı kriterlere dikkat etmek gerekmektedir. Akvaryumdaki diğer etkenlerin seçiminde ve dizaynında olduğu gibi aydınlatmanın seçiminde de akvaryumun uyumu göz önüne alınması gerekmektedir.

Yani akvaryumdaki bitkilerin ve balıkların benzer ışık özelliklerine ihtiyaç duyması gerekmektedir. Aksi takdirde aydınlatma balık veya bitki için seçilmek zorunda kalınır veya seçilmiş aydınlatma miktarı ve tipi, bitki veya balığı rahatsız edici olabilir.

### 2.3.3.1. Balık aydınlatması

Balıkların doğal ortamlarındaki ışık faktörlerinin sağlanması için yapılır. Eğer balık bir yüzeye yakın ve bol ışık altında yaşayan bir balık ise olabildiğince güneş ışığına yakın ve yüksek şiddette aydınlatma kullanılabilir. Eğer balık derinde veya loş ortamda yaşayan balık türü ise aydınlatma bu ambiyansı yakalayacak şekilde olmalıdır. Daha çok estetik aydınlatmadaki hususlar göz önüne alınır. Örneğin; Piranha'ların veya bolca Loach türlerinin bulunduğu şiddetli ve yüksek kelvinli (Daylight-güneş ışığı) aydınlatma uygulamak, balıkların stres ve huzursuzluktan hastalanmalarına neden olur.

### 2.3.3.2. Bitki aydınlatması

Bitkilerin büyümesi için ışık, gübre, iyi bir dip kumu, uygun akvaryum ortamı vs.den çok daha fazla gereklidir. Bitkiler fotosentez yaparlar ve bu sayede yaşamlarını sürdürürler (Diğer bütün faktörler de çok önemlidir). Fotosentez yapılabilmesi için bilindiği gibi en önemli faktör ışıktır. Fotosentez sadece ışık altında olur ve ortamdan bitkiler CO<sub>2</sub> (karbondioksit) çekerek yerine O<sub>2</sub> (oksijen) verirler. Işık kesildiğinde ise durum tam tersi şekilde cereyan eder ve bitki ortamdan (akvaryumdan) oksijen alıp ortama CO<sub>2</sub> verirler. Bitki aydınlatması bitkilerdeki fotosentez olayını arttırıcı olacak şekilde tasarlanması gerekmektedir. Işığın kalitesi de miktarı kadar önemlidir.

Yapılan araştırmalara göre fotosentez özellikle mor ötesi (ultraviyole) ve kırmızı 450 nanometre dalga boyu ve 650 nanometre dalga boyunda çok fazla artış göstermektedir. Fotosentez olayını arttırmak için güneş ışığı özellikleri gösteren lambaların seçimi ile birlikte bu tip özel lambaların seçimi bitkinin fotosentez yapmasında artışa sebep olacak ve bitkiler çok daha hızlı büyüyeceklerdir.

Balıkları fazla ışıktan korkutmadan çok aydınlatma gerekmektedir. Işık şiddetinin neden olabildiğince çok olması şöyle açıklanabilir. Öğle saatinde güneş ışığının şiddeti 100000 lüxe kadar çıkmaktadır (Kelvin ile karıştırılmasın. Çünkü ikisi de binli sayılarla ifade ediliyor) veya daha değişik bir şekilde anlatılacak olunursa öğlen

saatinde güneşe çıplak göz ile bakmaya çalışıldığında bu güneşin gücü ve olması gereken ışık şiddeti için ipucu verecektir.

### **2.3.4. Lamba tipleri**

#### **2.3.4.1. Elektrik ampulü**

Evlerde kullanılan ampuller ilk yatırım maliyeti düşük fakat genel olarak verimsiz lambalardır. Bu tip lambalardan çıkan ışığın rengi 2800 K (Kelvin) den aşağıda ve kırmızı dalga boyuna yakındır (yani bol ısı üretir). Genellikle düşük verimli olmaları ve yaydıkları ışık renkleri (spektrum) açısından çok fazla tavsiye edilmez. Sadece eğer akvaryumu aydınlatan diğer lambaların ışık spektrumunda kırmızı ve kırmızı ötesi renkler bulunmuyorsa ek lamba olarak kullanılabilir. (Mesela cool white floresan ile birlikte). Halojen ile birlikte renk renderleme faktörü yüksek olduğundan dolayı estetik aydınlatmada kullanılabilir. Çıkardıkları sıcaklığa dikkat edilmelidir. Halojen lamba ile birlikte ömürleri diğer lambalara göre kısadır. (Ortalama 1000–2000 saat)

#### **2.3.4.2. Halojen lambalar**

Halojen lambalar ışık özellikleri ve tipi açısından ampullere benzer. Biraz daha verimlidirler. Yaydığı ışığın rengi 3000 Kelvin civarında olup daha beyazdır. İlk kurulum maliyetleri daha yüksektir. Bütün diğer özellikleri ampul ile aynıdır.

#### **2.3.4.3. Floresan lambalar**

Değişik ışık renklerinde, güç ve tipte bulunabilirler. Verimleri yüksektir. Çok az sıcaklık yaydıkları için akvaryumun sıcaklığını yükseltmezler. Ömürleri yüksektir. (Ortalama 10000-20000 saat arası) İlk kurulum maliyetleri fazla olsa da işletme maliyetleri daha düşüktür. Floresan seçimi yapılırken güneş ışığı veya sıcaklık derecesi 5000K veya daha yüksek sıcaklık seçimi yapılması tavsiye edilir. Bitkiler açısından kullanılması tavsiye edilebilecek lambalar şunlardır:

1. OSRAM Lumilux Plus: Kar beyazından sarımsı kreme kadar değişik renk



seçeneđi mevcut. Avrupalı akvaristlerce çok tavsiye edilen bir ampul. Kaliteli bir üretime sahiptir.

2. OSRAM Fluora: Geliştirilmiş gro-lux tarzı ampul diyebiliriz. Ev içi bitkileri için kullanılır. Akvaryumda diđer ampullerle beraber iyi sonuç verir.
3. Philips Aquarelle: Philipsin ürettiđi bu floresan özellikle akvaryumlar için tasarlanmış olup akvaryumda fotosentez için idealdir. Işıđın sıcaklığı: 10000 Kelvin.
4. Philips Agro-Lite: Philipsin ürettiđi bu floresan bitkilerde fotosentezi arttıracak şekilde tasarlanmıştır.
5. Philips Daylight F40D: Güneş ışığı tipindeki bu floresan 6500 Kelvin derecesinde ışık üretmektedir. Işık spektrumu bitkiler için uygundur.
6. Sylvania Gro-Lux: Sylvania tarafından üretilen bu lamba bitkiler için üretilmiştir. Ürettiđi ışık pembe olup Işık Renderleme faktörü çok düşüktür.
7. Sylvania DaylightStar: 5000K sıcaklıkta güneş ışığı. Yalnız tek başına yetersiz olabilir. Özellikle 10000K'lik tropik ışık gerektiren bitkilerde.
8. Sylvania AquaStar: tam 10000K'lik olup, tropik dik açılı gün ışığını taklit eder. Deniz akvaryumlarında da kullanılır. Kırmızı ve Mavi renkler daha fazla.
9. Philips ColorTone 50 veya 75: Bütün renklerin dengeli bir şekilde üretildiđi bu lamba 5000 Kelvin sıcaklığında ışık üretir.

General Elektrik'te güneş ışığı tipinde floresan lambalar üretmektedir. Bunun yanında sadece bitki ve akvaryum için Plant Light ve Plant & Aquarium tipinde floresanlar üretmektedir.

Dünyada sadece akvaryum için üretilmiş başka floresan lambalar bulunmaktadır. Fakat bunlar maalesef ülkemizde bulunmamaktadır.

Floresan lambalar belli bir kullanımdan sonra yaydıkları ışık miktarı azalır ve spektrumları deđişir. Bu sebepten dolayı belli sürelerde (mesela 6 ayda - 1 yılda bir) bu lambalar bozulmasa da deđiştirilmelidir. Floresan lambalar genellikle akvaryumla gelen lamba apareyi ile birlikte kullanılmaktadır. Bu aparey bir reflektör (ışığı yansıtıcı) içermez. Bu sebepten dolayı lambanın yaydığı ışık azalmakta ve lambadan elde edilebilecek fayda azalmaktadır. Bu apareylerde her zaman yetersiz ampul yeri vardır. Eninde sonunda en uygun aydınlatma için yeni bir kapak ve reflektör temin

edilmeli veya yapılmalıdır.

#### **2.3.4.4. Cıva ve sodyum buharlı lambalar**

Yüksek yoğunluklu deşarj tipi lambalardandır. Bu tür lambalar özellikle yüksek verimlerinden ve yüksek ışık şiddeti verebildiklerinden dolayı sokak lambalarında kullanılır. Yüksek sıcaklık yaymaları akvaryum suyunun sıcaklığının artmasına ve uygulama zorluğuna sebep olur. Su soğutucusu veya fan gerektirirler.

a) Cıva Buharlı Lambalar: Bu tür lambaların yaydığı ışık Maviye yakın güneş ışığı tipi floresanın yaydığı ışığa benzer. Işık spektrumu da çok az kırmızı ışık içerir. Ömürleri uzundur. İlk kurulum maliyetleri yüksektir. Lambaları pahalıdır. Cıva buharlı lamba yerine güneş ışığı tipi floresan kullanılması daha mantıklıdır.

b) Sodyum Buharlı Lambalar: Bu tip lambalar sarı ışık üretirler. Işık spektrumları fotosentezi desteleyecek yapıda değildir. Akvaryumda kullanılması tavsiye edilmez.

#### **2.3.4.5. Metal halide**

Metal Halide cıva ve sodyum buharlı lambalar gibi yüksek yoğunluklu deşarj tipi lambalardandır. Metal Halide yüksek verimli ve uzun ömürlüdür. Yüksek sıcaklık yaydıkları için uygulamada dikkat edilmelidir. Fan veya su soğutucusu gerektirebilirler. Değişik güçlerde bulunabilirler. En az 5000 Kelvin sıcaklıkta ışık üreten metal halid lambalar kullanılması tavsiye edilir. (Mesela Philips'in 5K High Output Metal Halide) Işık renderleme faktörleri çok kötüdür. Akvaryumlarda 250 Wattın altında metal halide kullanılması tavsiye edilmez.(Işık sıcaklığı ve verimleri düşmektedir) [6].

#### **2.4. Akvaryum Kumu**

Akvaryum kumu olarak genellikle taneleri 2-4mm kalınlığında kireçsiz dere kumu veya quarz kum kullanılır. Koyu renk tonlu kum tipleri akvaryumlarda daha dekoratif bir görünüm verirler. Akvaryum tabanında oluşturulacak kum örtüsünün

yaklaşık 5cm kalınlığında olması tavsiye edilir. Birçok biyolojik dönüşüm akvaryum tabanında yer alır. Bu dönüşümler sonucu ortaya çıkan organik maddelerin bir kısmı, bitki kökleri tarafından besin maddesi olarak kullanılmak üzere emilirler. Bazı balık türleri kumu kazarak kendilerine veya yavrularına yuva yaparlar. Akvaryum kumundaki su dolaşımı biyolojik denge için son derece önemlidir. Kumda çok hızlı su dolaşımı besin maddelerinin birçoğunun oksitlenerek çökmesine ve bitkiler tarafından kullanılmayacak bir hale gelmesine neden olur. Taban filtreleri bu yüzden bitkili akvaryumlarda tavsiye edilmezler.

Çok yavaş su dolaşımı ise kumda oksijenin çok azalmasına ve kokuşmaya neden olur. Ayrıca dolaşım olmazsa besin maddeleri bitki köklerine taşınmazlar. Çok ince kum kullanmak bu yüzden bitkiler için ideal değildir.

Son yıllarda kumda en uygun su dolaşımını sağlamak için düşük güçte taban ısıtıcıları kullanılıyor ve bitkiler, dolayısıyla balıklar üstünde de olumlu sonuçları da görüyor.

Laterit, tropik bölgelerde çok görülen, mineraller yönüyle (özellikle de demirce) zengin kırmızı bir kum tipidir. Akvaryum kurulurken kum konulmadan tabana bir miktar laterit dökülmesi bitkilerin uzun süreler demir eksikliği çekmeden büyüebilmesine yardım edecektir [7] .

## 2.5. Isıtıcı

Akvaryum balıkları genellikle yaklaşık 25 °C sıcaklığında suda yaşar. Bu nedenle akvaryum suyu doğru şekilde ısıtılmalıdır. Akvaryum ısıtıcısı akvaryuma suyun sürekli olarak etrafından akacağı ve eşit olarak ısınacağı şekilde yerleştirilmelidir.

Kaç wattlık bir cihaz kullanılması gerektiği kolaylıkla belirlenebilir: Az ısıtılan odalarda suyun litre başına yaklaşık 1,5watt olarak hesaplanır. Akvaryum sıcak bir odada bulunuyorsa litre başına 1watt yeterlidir. Hiç çekinilmeden daha kuvvetli bir ısıtıcı tercih edilebilir; belirli miktarda ısıyı üretmek için harcanılan güç aynıdır. Kuvvetli bir ısıtıcının güç rezervi daha büyük olduğundan evin ısıtma sistemi devre

dışı kalsa bile akvaryum gerekli şekilde ısınabilir. Çeşitli akvaryum boyları göre suyun sıcaklığını 5°C, 10°C ve 15°C artırmak için gerekli ısıtıcı güçleri tablo 2.2’de görülmektedir [8] .

Tablo 2.2. Akvaryum hacmine göre ısıtıcı güçleri

$\Delta T=T_2-T_1^*$		Akvaryum Hacmi							
Litre		25	50	75	100	150	200	250	300
$\Delta T$	5°C	25W	50W	50W	75W	100W	150W	200W	250W
	10°C	25W	50W	75W	100W	150W	200W	250W	300W
	15°C	75W	100W	150W	200W	300W	2* 200W	2* 250W	2* 300W

## 2.6. Hava Motoru

Akvaryuma oksijen için gereken havayı basar, hava ile çalışan filtreleri besler, akvaryumda suyun hareket edemediği ölü noktalar var ise hava taşları bu bölgelere konularak buralarda da sirkülasyonu sağlar. 50 l/h – 80 l/h – 100 l/h – 200 l/h gibi çeşitli güçte olanları mevcuttur. Akvaryum büyüklüğüne göre ve sessiz olanları tercih edilmelidir. [3]

## 2.7. Filtreler ve Filtreleme

### 2.7.1. Neden filtre?

Balıklar solungaçları ve dışkıları vasıtasıyla amonyak üretirler. Ayrıca amonyak, akvaryumda bulunan bozulmuş yem artıkları ve organik bileşiklerin aracılığıyla da meydana gelir. Amonyak balıklar için zehirlidir. Bu maddenin yüksek olduğu akvaryumlardaki balıkların ani ölümleri kaçınılmaz olmakla beraber amonyağın daha az olduğu ortamda uzun süre yaşayan balıklar da tehlike altındadır.

Amonyak balığı direkt olarak solungaçlarını yakarak öldürmese bile hayvanın direncini kırarak akvaryum içinde bulunan zararlı bakteri ve organizmalara balığı savunmasız bırakır. Suyun ph derecesi asitleştikçe sudaki amonyak balık için daha zararsız hale gelecektir aynı zamanda ph yükseldikçe amonyak balığı zehirleyecek seviyelere çıkar.

Peki, bununla ilgili ne yapılabilir? Amonyacı sıfırlamak veya sudan arıtmak gerekmektedir. Amonyak Amquel veya zeolit kil'i gibi kimyasallar ile yok edilebilir. Ancak bunlar kısa dönem için koruyucudurlar, yenilenmedikleri takdirde amonyak yeniden yükselecektir. Bu yöntem sizin için pahalı olacaktır. Ayrıca suyun niteliğini bozan diğer maddeler de vardır, balıklar için tek tehlike amonyak değildir ve bu kimyasallar sadece amonyağın etkilerini zararsız hale getirdikleri için sadece amonyağın yükseldiği durumlarda gereklidirler.

Her gün düzenli şekilde su değiştirilebilir. Ancak birçok kişi bunu haftalık periyotlarla yapmaktadır ve bu şekilde fazla problem yaşanmamaktadır.

Sonuç olarak anlaşılacağı üzere tek çözüm filtrelemedir. Filtreler 3 kategoride suyu temizlerler. Bunlar biyolojik, mekanik ve kimyasal filtrasyondur.

### **2.7.1.1. Biyolojik filtrasyon**

Filtreler birçok faydalı bakteri kolonisine oksijence zengin bir ortam oluşturur. Bunlardan bir bölümü nitriti daha az zehirli bir madde olan nitrate dönüştürür. (Ancak nitrat seviyesi arttıkça balıkların büyümeleri yavaşlar hatta durur) Nitrat düzenli su değişimleriyle ve nitratı yok eden kimyasallar vasıtasıyla sudan azaltılmalıdır.

Etkili bir biyolojik filtreleme için bakteriler için amonyak ve oksijenin bulunduğu bir ortam sağlamalıdır. Alanın çokluğu daha fazla bakteri kolonisi oluşmasına katkıda bulunacaktır.

### 2.7.1.2. Mekanik filtrasyon

Filtreler katı balık atıklarını, yenmemiş yemleri ve organik parçaları akvaryumdan uzaklaştırırlar. Eğer mekanik filtrasyon malzemeleri düzenli olarak yıkanır (kloruz ve normal sıcaklıkta su ile) bu atık maddeler amonyağa dönüşmeden önce akvaryumdan atılmış olur.

Herhangi gözenekli malzeme mekanik filtrasyon malzemesi olarak kullanılabilir ama sünger, kağıt kartuşlar, polyester petler daha iyi iş görür. Eğer gözenekler çok ufak ise çabuk tıkanır, eğer çok büyük olurlarsa tutmaları gereken maddeleri tutamazlar. Etkili bir mekanik filtrasyonun birinci şartı, filtre malzemelerinin düzenli temizlenmesi veya değiştirilmesidir. Çoğu akvaryum sever bundan kaçındıkları için gözeneklerin tıkanması sonucunda amonyak zehirlenmesinden dolayı balıklarını kaybederler.

### 2.7.1.3. Kimyasal Filtrasyon

Kimyasal açıdan etkili filtre malzemeleri amonyağı veya potansiyel zehir olan çözülmüş atıkları nötrlemek veya uzaklaştırmak için filtreye konulur. Ancak kapasitesini dolduran kimyasal malzeme içinde tuttuğu toksin etkisi olan maddeleri suya geri bırakır bu yüzden belli zaman aralıkları ile değiştirilmelidir.

En çok kullanılan kimyasal filtre malzemesi aktif karbondur. Ayda bir değiştirilen ve 20 galona (yaklaşık 80 litre) yarım kup (büyük su bardağı) ölçekte kullanılan karbon en iyi sonucu verecektir. Kimyasal filtre malzemeleri öngörülen zaman aralıklarında değiştirilmelidir aksi takdirde bünyelerinde tuttukları toksin maddeleri suya geri verirler.

## 2.7.2. Filtre çeşitleri

### 2.7.2.1. Sünger filtreler

Bu tip filtreler hava motoruna bağlı olarak çalışırlar. Genelde küçük hacimli (60

litreye kadar) akvaryumlarda özellikle üretim akvaryumlarında kullanılırlar. Kullanımı rahat, ekonomik ve etkili bir biyolojik filtre malzemesidir. Kullanıldıkları akvaryumdan yeni kurulmuş bir akvaryuma konularak biyolojik döngüyü başlatabilirler. Ayrıca mekanik filtreleme de yaparlar dolayısı ile bu süngerler etkili olabilmeleri için her ay klorsuz ve oda sıcaklığında su ile çalkalanarak temizlenmelidirler.

Bu filtreler yalnızca küçük akvaryumlarda kullanılmalıdır. İçerisinde fazla miktarda balık bulunan akvaryumlarda pek işe yaramazlar. Uygun akvaryumlarda bitkilerin, taşların veya ağaç köklerinin arkalarına gizlenebilirler.

Eğer sünger filtre kullanılıyorsa her 3 günde bir veya haftada bir dipte birikmiş olan kaba kirleri bir hortum vasıtasıyla akvaryumdan uzaklaştırılmalıdır ve yerine klordan arındırılmış ve akvaryumdaki mevcut su ile aynı koşullara getirilmiş su eklenmelidir.

### **2.7.2.2. Kutu filtreler**

Kutu filtreler sünger filtrelerin daha gelişmiş ve etkili bir versiyonudur. Bünyelerinde kendi motorları bulunur ve sünger filtreden en büyük getirileri içerilerine değişik filtre malzemelerin (karbon, amonyak emici vs.) koyulabilmesidir. Bu filtreler de sünger tip gibi ucuzdur ve kolayca başka bir akvaryuma nakledilebilirler.

### **2.7.2.3. Kumaltı filtreleri (taban filtresi)**

Kumaltı filtreleri plastik tabakalar halinde kumun altına yerleştirilir. Su plastik tabakalar arasındaki gözeneklerden geçerek sürekli olarak temiz kalır. Genelde akvaryumseverler kumun taban malzemesi arasına kaçıp kaçmadığını merak eder. Kum bu tabakanın üzerinde kalır ve içine kaçması söz konusu değildir. Dipten çekilen su yüzeyden tekrar temizlenmiş olarak akvaryuma geri verilir.

Bu tip filtrelemenin görünmeme gibi estetik avantajları olsa da temizlemesi zordur, ilaçlama yapıldığında üzerindeki biyolojik hayat ölür, bitki yetişmesine uygun değildir.

#### 2.7.2.4. Motorlu filtreler

Bu tip filtreler akvaryumun arka veya yan camına asılarak kullanılır. Güçlü şekilde mekanik, biyolojik ve kimyasal filtreleme yapabilirler. Suyu çok kirleten balıklar veya kalabalık akvaryumlar için kullanılırlar.

Çoğunlukla motorlu filtreler mekanik (sünger, elyaf), biyolojik (sünger, biyolojik taşlar vs.) ve kimyasal (karbon ve zeolit) malzemelerle beraber satılırlar. Bu çeşitlilik etkili bir filtrasyon sağlar.

Örneğin alkalın ( $\text{ph}>7$ ) ve sert (karbonat sertliği  $>15$ ) su isteyen afrika çiklitleri için bu tip filtrenin içerisine mercan kumu veya mercan parçaları konulur. Çünkü mercanlar iskeletlerini kalsiyum karbonattan oluştururlar ve ölü mercanlar su akıntısı içerisinde erime eğilimindedirler. Dolayısı ile konuldukları filtre içerisinde suyun karbonat sertliğini yükselterek afrika balıklarının istediği suyu elde etmemize yardımcı olurken sabit bir ph derecesi de sağlar.



Şekil 2.1. Dış filtre

Motorlu filtreler bu avantajların yanında gürültülü çalışma, göz estetiğini bozma ve açık ortamda bulunmalarından dolayı içerisindeki suyun buharlaşma dezavantajlarına sahiptir. Şimdi daha kullanışlı olan kanister filtreleri (dış filtreler) inceleyelim. Bu



filtreler motorlu filtrelerin geliştirilmiş ve en çok kullanılan tipidir. Bunlar akvaryumun dışında durur ve sistemleri bir kova ihtiva eder. Filtre malzemeleri bu kovanın içerisinde bulunur. Su girişi ve çıkışı için 2 ayrı hortumu vardır (Şekil 2.1). Motorlu (power) filtrelerden daha kuvvetli ve daha kullanışlıdır. Akvaryumdan yüksek bir hızda tutulurlarsa su kaçırma problemi hallolmuş olur.

Kanisterler oldukça sessiz çalışırlar. En bilinen markalar Eheim ve Sacem 'dir. Bu filtrelerin avantajları kapasiteleridir 1000 litreye kadar olan versiyonları vardır. Ancak bunların dezavantajları ise temizlenmeleri zordur, pahalıdır ve temizlenmedikleri takdirde yüksek miktarda nitrat açığa çıkarırlar.

### **2.7.3. Nitrat önleyiciler**

#### **2.7.3.1. Kum filtreleri**

Bu tip filtreler suyun bir kum katmanından yavaşça geçirilerek temizlenmesi yoluyla yapılırlar. Yavaş akan su ve geniş alan, aerobik (yararlı bakteriler) bakteriler için uygun yaşam alanıdır. Sudaki nitratı aerobik bakteriler, su kumun alt katmanlarına geçmeden yiyecek olarak tüketirler. Ancak dikkat edilmesi gereken koşul suyun yavaş akmasıdır aksi halde suya nitrat bakteriler tarafından yok edilemeden geri verilmiş olur.

#### **2.7.3.2. Bitkiler**

Bitkiler de aerobik bakteriler gibi nitratı besin olarak kullanırlar ancak çürüyen kısımları suya amonyak vereceğinden bu kısımlar iyice temizlenmeli ve yerine taze su konulmalıdır. İyi bitkilendirilmiş bir akvaryumun bakımı biraz zordur. İyi ışık ve bitkileri besleyecek kaliteli taban malzemeleri ve düzenli gübreleme gerekir.

### **2.7.4. Filtrelemede unutulmaması gerekenler**

Her ne kadar kuvvetli bir filtrasyon yapılsa da haftalık dipten pislik çekme ve yerine temiz, klordan arındırılmış taze su eklenmelidir. Haftalık en az %20 su değişimi

şarttır.

Özellikle sünger filtre kullanılan akvaryumlarda su deęişimi daha yoğun (2–3 günde bir) yapılmalıdır.

Tüm filtre malzemelerini üretici firmanın öngördüğü zaman periyotlarında temizlenmeli ve deęiştirilmelidir. Aksi halde bu malzemelerin barındırdıkları pislik ve toksin maddeleri suya geri vereceęi unutmamalıdır.

Akvaryumun hacmine uygun filtre kullanılmalıdır.

Kanister filtrelerin sürekli çalıştığına emin olunmalıdır (gece dahi). Aksi halde içerisindeki yararlı bakterilerin ölmeye başladıkları akıldan çıkarılmamalıdır [9] .

## **BÖLÜM 3. PIC 16F877 MİKRODENETLEYİCİ**

### **3.1. PIC Nedir?**

PIC serisi mikroişlemciler MICROCHIP firması tarafından geliştirilmiş ve üretim amacı çok fonksiyonlu mantıksal uygulamaların hızlı ve ucuz bir mikroişlemci ile yazılım yoluyla karşılanmasıdır. PIC'in kelime anlamı – Peripheral Interface Controller - giriş çıkış işlemcisidir. İlk olarak 1994 yılında 16 bitlik ve 32 bitlik büyük işlemcilerin giriş ve çıkışlarındaki yükü azaltmak ve denetlemek amacıyla çok hızlı ve ucuz bir çözüme ihtiyaç duyulduğu için geliştirilmiştir. Çok geniş bir ürün ailesinin ilk üyesi olan PIC16C54 bu ihtiyacın ilk meyvesidir. PIC işlemcileri RISC benzeri işlemciler olarak anılır. PIC16C54 12 bit komut hafıza genişliği olan 8 bitlik CMOS bir işlemcidir. 18 bacaklı dip kılıfta 13 G / Ç bacağına sahiptir, 20 MHz osilator hızına kadar kullanılabilir ve 33 adet komut içermektedir. 512 byte program EPROM'u ve 25 byte RAM'i bulunmaktadır. Bu hafıza kapasitesi ilk bakışta çok yetersiz gelebilir ama bir RISC işlemci olması birçok işlevin bu kapasitede uygulanmasına olanak vermektedir. PIC serisi tüm işlemciler herhangi bir ek bellek veya giriş / çıkış elemanı gerektirmeden sadece 2 adet kondansatör, 1 adet direnç ve bir kristal ile çalıştırılabilmektedir. Tek bacadan 40 mA akım çekilebilmekte ve tümdevre toplamı olarak 150 mA akım akıtma kapasitesine sahiptir. Tüm devrenin 4 MHz osilator frekansında çektiği akım çalışırken 2 mA, bekleme durumunda ise 20uA kadardır. PIC 16C54'ün fiyatının yaklaşık 2.0 Amerikan Doları civarında olduğu düşünülürse bu işlemcinin avantajı kolayca anlaşılır. PIC 16C54 'un mensup olduğu işlemci ailesi 12Bit Core 16C5X olarak anılır. Bu gruba temel grup adı verilir. Bu ailenin üyesi diğer işlemciler PIC16C57, PIC16C58 ve dünyanın en küçük işlemcisi olarak anılan 8 bacaklı PIC12C508 ve PIC 12C509'dur. Interrupt kapasitesi ilk işlemci ailesi olan 12Bit Core 16C5X ailesinde bulunmamaktadır. Daha sonra üretilen ve orta sınıf olarak tanınan 14Bit Core- 16CXX ailesi birçok açıdan daha yetenekli bir grup işlemcidir. Bu ailenin temel özelliği interrupt kapasitesi ve 14 bitlik komut işleme hafızasıdır. Bu özellikler PIC'i gerçek bir işlemci olmaya ve

karmaşık işlemlerde kullanılmaya yatkın hale getirmiştir. PIC16CXX ailesi en geniş ürün yelpazesine sahip ailedir. 16CXX ailesinin en önemli özellikleri seri olarak devre üstünde dahi programlanması -ki bu özellik PIC16C5x de epey karmaşıktır, paralel programlanabiliyordu- interrupt kabul edebilmesi, 33 G / Ç, A/S Çevirici, USART, I2C, SPI gibi endüstri standardı giriş çıkışları kabul edecek işlemciler ürün yelpazesinde yer vermesidir. PIC 16CXX ailesinin amatör elektronikçiler arasında en çok tanınan ve dünyada üzerinde en çok proje üretilmiş, internetin gözdesi olan bireyi PIC16C84 veya yeni adıyla PIC16F84 dur. PIC 16F84 un bu kadar popüler olması onun çok iyi bir işlemci olmasından ziyade program belleğinin EEPROM - Elektrikle silinip yazılabilen bellek - olmasından kaynaklanmaktadır. Seri olarak dört adet kabloyla programlanması da diğer önemli avantajıdır. Bugüne kadar amatörce bir işlemciyle uğraşmış herkesin en büyük sıkıntısı EPROM veya EPROM tabanlı işlemcileri programladıktan sonra morötesi ışık kaynağı ile silip tekrar programlamaktır. Bu çok zahmetli ve bir amatör için ekipman gerektiren yöntem olmuştur. Evde üretilmesi zor olan özel bir programlayıcı da madalyonun diğer yüzüdür. PIC16F84, amatörler tarafından internette en bol programlayıcısı bulunan işlemcilerden biridir. EPROM silmek diye bir kavram zaten söz konusu değildir ayrıca PIC16F84 EEPROM barındırdığından mümkün değildir. EEPROM belleği programlayan programlayıcı devre 1 saniye içinde aynı belleği silebilmektedir. Bu özellik tasarımcıya çok hızlı ve defalarca deneyerek program geliştirme avantajını getirmektedir. Bu tasarımcı için çok önemli ve gerekli bir özelliktir. Benzetim programları genel fikir vermek açısından çok yararlı olsalar da ciddi tasarımlarda devreyi fiziksel olarak gerçeklemek, deneyler yapmak kesinlikle gereklidir. Bu denemeleri yaparken işlemciyi devrenizden sökmek dahi gerekmez. Bu tip programlamaya ISP - In System Programming - denmektedir.

### 3.2. Neden PIC?

Tablo 3.1. Neden PIC?

Güvenilirlik	Tüm komutlar 12 veya 14 bit'lik bir program bellek sözcüğüne sığar. Yazılımın, programın veri kısmına atlamaya ve veriyi komut gibi çalıştırmasına gerek yoktur. Bu 8 bit'lik bus kullanan ve Harvard mimarisi temelli olmayan mikrodenetleyicilerde gerçekleşmektedir.
--------------	---

Tablo 3.1. (Devam) Neden PIC?

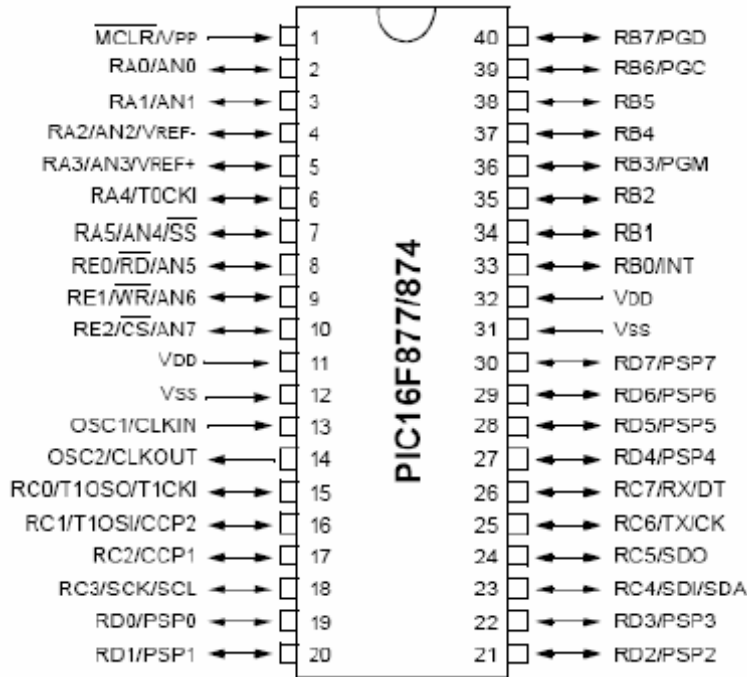
Statik İşlem	PIC tamamıyla statik bir mikroişlemcidir. Başka bir deyişle saati durdurduğunuzda, tüm yazmaç içeriği korunur. Pratikte bunu tam olarak gerçekleştirmeniz mümkün değildir. PIC' i uyutma moduna getirdiğinizde, saat durur ve PIC' e uyutma işleminden önce hangi durumda olduğunu size hatırlatacak çeşitli bayraklar kurar. PIC uyuma modunda yalnızca 1 mA'dan küçük değere sahip bekleme akımı çeker.
Hız	PIC, osilatör ve yerleşik saat yolu arasına bağlı yerleşik bir 4' lü bölünme' ye sahiptir. Bu, özellikle 4 MHz'lik kristal kullanıldığında komut sürelerinin hesaplanmasında kolaylık sağlar. Her bir komut döngüsü 1 ms' dir. PIC oldukça hızlı bir mikrodur. Örneğin 5 milyon komutluk bir programın, 20 MHz'lik bir kristalle adımlanması yalnız 1 saniye sürer. Bu süre 386 SX 33 işlemcisinin hızının neredeyse 2 katıdır.
Kod Verimliliği	PIC, Harvard mimarisi temelli 8 bit' lik bir mikrodenetleyicidir. Bu, bellek ve veri için ayrı yerleşik bus'ların bulunduğu anlamına gelir. Böylelikle akış miktarı veriye ve program belleğine anında erişim sayesinde arttırılmış olur. Geleneksel mikrodenetleyicilerde veri ve programı taşıyan bir tek yerleşik bus bulunur. Bu, PIC ile karşılaştırıldığında işlem hızını en az iki kat yavaşlatır.
Komut Seti	16C5x ailesinde yazılımı yaratmanız için 33 komut öğrenmeniz yeterlidir. 16Cxx araçları içinse bu sayı 35'tir. PIC tarafından kullanılan komutların hepsi yazmaç temellidir ve 16C5x ailesinde 12 bit 16Cxx ailesindeyse 14 bit uzunluğundadır. CALL, GOTO ve bit test eden BTFSS, INCFSZ gibi komutlar dışında, her bir komut, tek bir çevrimde çalışır. Mikrodenetleyicinin çalışmasını ve işletmesini sağlayan bilgidir. Başarılı bir uygulama veya ürün isteniyorsa yazılım hatasız olmalıdır. Yazılım C, BASIC veya Assembler gibi çeşitli dillerde veya ikilik (binary) olarak yazılabilir.

### 3.3. PIC16F877 özellikleri ve yapısı

PIC16F877, belki en popüler PIC işlemcisi olan PIC16F84'ten sonra kullanıcılarına yeni ve gelişmiş olanaklar sunmasıyla hemen göze çarpmaktadır. Program belleği FLASH ROM olan PIC16F877'de, yüklenen program PIC16F84'te olduğu gibi elektriksel olarak silinip yeniden yüklenebilmektedir. Tablo 3.2'de PIC16F877 ve PIC16F84 işlemcileri arasında özellik karşılaştırması yapılmıştır. Özellikle PIC16C6X ve PIC16C7X ailesinin tüm özelliklerini barındırması, PIC16F877'yi kod geliştirmede de ideal bir çözüm olarak gündeme getirmektedir. Konfigürasyon bitlerine dikkat etmek şartıyla C6X veya C7X ailesinden herhangi bir işlemci için geliştirilen kod hemen hiçbir değişikliğe tabi tutmadan F877'e yüklenebilir ve çalışmalarda denenebilir. Bunun yanı sıra PIC16F877, PIC16C74 ve PIC16C77 işlemcileriyle de bire bir bacak uyumludur. PIC16F877'nin bacak tanımlamaları şekil 3.1'de verilmiştir [10].

Tablo 3.2. PIC16F877 ve PIC16F84 işlemcilerinin kıyaslanması

ÖZELLİKLER	PIC 16F877	PIC 16F84
Çalışma Hızı	DC20MHz	DC10MHz
Program Belleği	8 K word Flash ROM	1K word flash ROM
EEPROM Belleği	256 Byte	64 Byte
Kullanıcı RAM	368 x 8 Byte	68 x 8 Byte
Giriş/Çıkış Port Sayısı	33	13
Zamanlayıcı	Timer0, Timer1, Timer2	Timer0
A/D Çevirici	8 Kanal, 10 Bit	Yok
Capture/Compare /PWM	16 Bit Capture, 16 Bit Compare, 10 Bit PWM çözünürlük	Yok
Seri çevresel Arayüz	SPI (Master) ve I2C (Master /Slave) modunda SPI portu (senkron seri port)	Yok
Paralel Uydu Port	8 Bit, harici, RD, WR ve CS kontrollü	Yok
USART/SCI	9 Bit adresli	Yok



Şekil 3.1. PIC16F877 Bacak tanımlamaları

16F87X Mikrodenetleyici ailesi aşağıdaki temel özellikleri taşır.

1. CPU azaltılmış komut seti
2. RISC temeline dayanır.
3. Öğrenilecek 35 komut vardır ve her biri 14 bit uzunluktadır.
4. Dallonma komutları iki çevrim (cycle) sürede, diğerleri ise bir çevrimlik sürede uygulanır.
5. İşlem hızı 16F877'de DC20 MHz'dir. (16F877'de bir komut DC200ns hızında çalışır.)
6. Veri yolu (databus) 8 bittir.
7. 32 adet SFR (Special Function Register) olarak adlandırılan özel işlem kaydedicisi vardır ve bunlar statik RAM üzerindedir.
8. 8 Kword'e kadar artan flash belleği 1 milyon kez programlanabilir.
9. 368 Byte'a kadar artan veri belleği (RAM),
10. 256 Byte'a kadar artan EEPROM veri belleği vardır.
11. Pin çıkışları PIC 16C73B/74B/76 ve 77 ile uyumludur.
12. 14 kaynaktan kesme yapabilir.
13. Yığın derinliği 8'dir.
14. Doğrudan, dolaylı ve göreceli adresleme yapabilir.

15. Poweron Reset (Enerji verildiğinde sistemi resetleme özelliği)
16. Powerup Timer (Powerup zamanlayıcı)
17. Osilatör Startup Timer (Osilatör başlatma zamanlayıcısı)
18. Watchdog Timer (Özel tip zamanlayıcı), devre içi RC osilatör
19. Programla kod güvenliğinin sağlanabilmesi özelliği
20. Devre içi Debugger (Hata ayıklamakta kullanılacak modül)
21. Düşük gerilimli programlama
22. Flash ROM program belleği (EEPROM özellikli program belleği)
23. Enerji tasarrufu sağlayan, uyku –Sleep Modu
24. Seçimli osilatör özellikleri
25. Düşük güçle, yüksek hızla erişilebilen, CMOSFlash EEPROM teknoloji
26. Tümüyle statik tasarım
27. 2 pinle programlanabilme özelliği
28. Yalnız 5V girişle, devre içi seri programlanabilme özelliği
29. İşleciminin program belleğine, okuma/yazma özelliği ile erişimi
30. 2.0 V – 5.0 V arasında değişen geniş işletim aralığı
31. 25 mA'lık kaynak akımı
32. Devre içi, iki pin ile hata ayıklama özelliği
33. Geniş sıcaklık aralığında çalışabilme özelliği
34. Düşük güçle çalışabilme özelliği

Çevresel özellikleri ise şöyle sıralanabilir:

1. TMR0: 8 bitlik zamanlayıcı, 8 bit önbölücülü
2. TMR1: Önbölücülü, 16 bit zamanlayıcı, uyuma modundayken dış kristal zamanlayıcıdan kontrolü arttırılabilir.
3. TMR2: 8 bitlik zamanlayıcı, hem önbölücü hem de sonbölücü sabiti
4. İki Capture / Compare / PWM modülü
5. 10 bit çok kanallı A/D çevirici
6. Senkron seri port (SSP), SPI (Master mod) ve I 2 C (Master Slave) ile birlikte
7. Paralel Slave Port, 8 bit genişlikte ve dış RD, WR, CS kontrolleri
8. USART/SCI, 9 bit adres yakalamalı
9. BOR Reset (Brown Out Reset) özelliği [11]



### 3.4. PIC16F877'nin besleme uçları ve beslenmesi

PIC16F877'nin besleme gerilimi 11, 12 ve 31, 32 numaralı bacaklardan uygulanmaktadır. 11 ve 32 numaralı Vdd ucu +5 V'a ve 12, 31 numaralı Vss ucu toprağa bağlanır. PIC'e ilk defa enerji verildiği anda meydana gelebilecek gerilim dalgalanmaları nedeniyle, oluşabilecek istenmeyen arızaları önlemek amacıyla 100nF'lık dekaplaj kondansatörünün devreye bağlanması gerekmektedir. PIC'ler CMOS teknolojisi ile üretildiklerinden 2 ila 6 Volt arasında çalışabilmektedirler. +5V'luk bir gerilim ise ideal bir değer olmaktadır.

### 3.5. PIC16F877'nin reset uçları

Kullanıcının programı kasti olarak kesip başlangıca döndürebilmesi için PIC'in 1 numaralı ucu MCLR olarak kullanılmaktadır. MCLR ucuna 0 Volt uygulandığında programın çalışması başlangıç adresine döner. Programın ilk başlangıç adresinden itibaren tekrar çalışabilmesi için, aynı uca +5 Volt gerilim uygulanmalıdır.

### 3.6. PIC16F877'nin saat uçları ve osilatör tipleri

PIC16CXX mikrodenetleyicilerinde 4 çeşit osilatör bulunmaktadır. Kullanıcı bu 4 çeşitten birini seçerek iki yapılandırma bitini (FOSC1 ve FOSC2) programlayabilir. Bu osilatör çeşitleri tablo 3.3'te verilmiştir. PIC16F877'de saat uçları 13. ve 14. bacaklardır. Hazırlanacak olan PIC programlarında kullanılan osilatör tipi PIC programının çalışma hızını ve hassasiyetini etkileyeceğinden dolayı amaca uygun bir osilatör devresi kullanılmalıdır. Tablo 3.3'te farklı osilatör çeşitleri ve özellikleri görülmektedir. Osilatör tipinin seçiminde dikkat edilecek bir başka nokta ise, seçilecek olan osilatörün kullanılan PIC'in özelliğine uygun olmasıdır. Örnek verecek olursak; en fazla 10 MHz çalışma frekansına sahip bir PIC16F877 için 20 MHz'lik bir osilatör kullanmak doğru olmaz. Fakat daha düşük bir frekans değeri ile çalışan bir osilatör devresi kullanılabilir.

Tablo 3.3. Osilatör çeşitleri

Osilatör Tipi	Tanımı	Özelliği	Frekansı
LP	Kristal osilatör veya seramik rezonatör	Asgari akım	40KHz
XT	Kristal osilatör veya seramik rezonatör	Genel amaçlı	4MHz
HS	Kristal osilatör veya seramik rezonatör	Yüksek hız	20MHz
RC	Direnç / Kapasitör zaman sabitli	Düşük maliyet	4MHz

### 3.6.1 Kristal osilatör / Seramik rezonatör

XT, LP ve HS modları, RC osilatörlere nazaran çok daha hassastırlar. Bu modlar, kristal osilatör veya rezonatörlerin, OSC1 / CLKIN ve OSC2 / CLKOUT uçlarına bağlanmalarıyla kurulmaktadır. Tablo 3.4'te hangi frekansta kaç pF'lık kondansatör kullanılması gerektiği belirtilmiştir.

Tablo 3.4 Frekansa göre kondansatör seçimi

OSİLATÖR TİPİ	FREKANS	KONDANSATÖR
LP	32 KHz	33-68 pF
LP	200 KHz	15-47 pF
LP	100 KHz	47-100pF
XT	500 KHz	20-68 pF
XT	1 MHz	15-68 pF
XT	2 MHz	15-47 pF
XT	4 MHz	15-33 pF
HS	8 MHz	15-47 pF
HS	20 MHz	15-47 pF

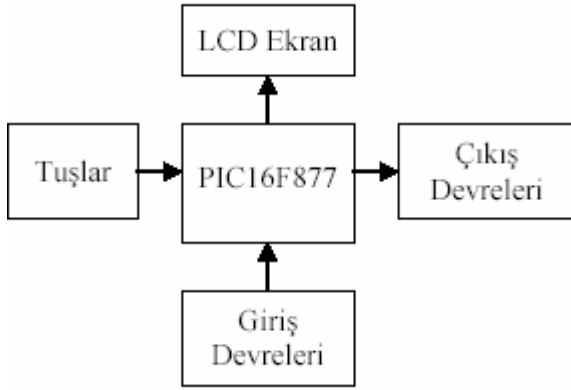
### 3.6.2. RC Osilatör

Zamanlamanın çok hassas olmadığı durumlarda RC ikilisi osilatör kaynağı olarak kullanılmaktadır. RC osilatör, maliyetin azaltılmasını sağlamaktadır. Kullanıcı dış R ve C elemanlarının toleransı nedeniyle meydana gelen değişiklikleri de dikkate

almalıdır. Direncin değeri 3 ila 100 KOhm arasında seçilmelidir. 1MOhm gibi yüksek direnç değerleri osilatörü gürültü ve nem gibi çevresel etkilere karşı duyarlı hale getirir. Direnç 2 KOhm değerinin altında ise, osilatör kararsız hale gelebilmekte, hatta tamamıyla durabilmektedir [10] .

## BÖLÜM 4. AKVARYUM OTOMASYONU TASARIMI

Akvaryum otomasyonunun genel donanım özelliklerini içeren temel blok şema şekil 4.1’de verilmiştir. Şekilden de görüldüğü gibi sistemin merkezinde kontrol işlemlerini gerçekleştirmek üzere PIC16F877 mikrodenetleyicisi bulunmaktadır. Tuşlar ve LCD ekran yardımıyla kullanıcı sistemi rahatlıkla kontrol edebilmektedir [12] .



Şekil 4.1. Sistemin blok şeması

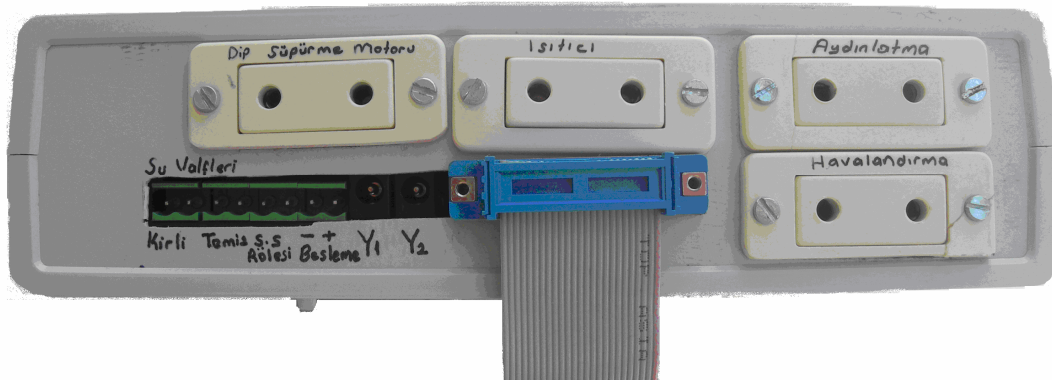
Akvaryum otomasyonunun yapabildikleri şöyle sıralanabilir.

1. Biri yetişkin biri yavru balıklar için hazırlanan iki yemlikten günün istenilen saatlerinde yemleme işlemi yapılabilmektedir.
2. Akvaryumun haftalık su değişimini otomatik olarak yapabilmektedir.
3. Ayrıca tasarımda bulunan göstergeli kontrol paneli sayesinde;
  - a. Sistem için gerekli saat ve tarih ayarları yapılabilmektedir.
  - b. Yemliklerin günün hangi saatinde yem atacakları ayarı yapılabilmektedir.
  - c. Su değişiminin gün ve saat ayarı yapılabilmektedir.
  - d. Yem atma, havalandırma ve aydınlatma işlemleri tek tuşla yapılabilmektedir.
4. Yem atma, havalandırma ve aydınlatma işlemleri uzaktan kumanda ile de yapılabilmektedir.
5. Su değiştirme mekanizması direkt olarak temiz ve pis su şebekesine

bağlanarak, bu sayede sisteme müdahale edilmeden daha uzun süreli bir otomasyon sağlanmıştır.

6. Enerji kesilmelerinde, kesintisiz güç kaynağı ile sistem desteklenip havalandırma ve yemleme yapılabilmektedir.

Bütün bu işlemler için gerekli olan yazılım ekler kısmında verilmiştir.



Şekil 4.2. Anakart devre kutusunun önden görünüşü

Yapılan akvaryum otomasyonunun elektronik kısmı iki bölümden oluşmaktadır. Birincisi PIC 16F877 mikrodenetleyicili anakart devresi, ikincisi göstergeli kontrol paneli devresidir. Anakart devre kutusunun önden görünüşü şekil 4.2’de, arkadan görünüşü şekil 4.3’de ve göstergeli kontrol paneli şekil 4.4’de görülmektedir.



Şekil 4.3. Anakart devre kutusunun arkadan görünüşü

Göstergeli kontrol panelinde, 4x16 LCD ekran, IR (uzaktan kumanda) alıcı, led (uzaktan kumanda uyarı ışığı), buzzer (ikaz hoparlörü), ısı sensörü (LM 35), LDR (ışık sensörü) ve tuş takımı için butonlar kullanılmıştır.

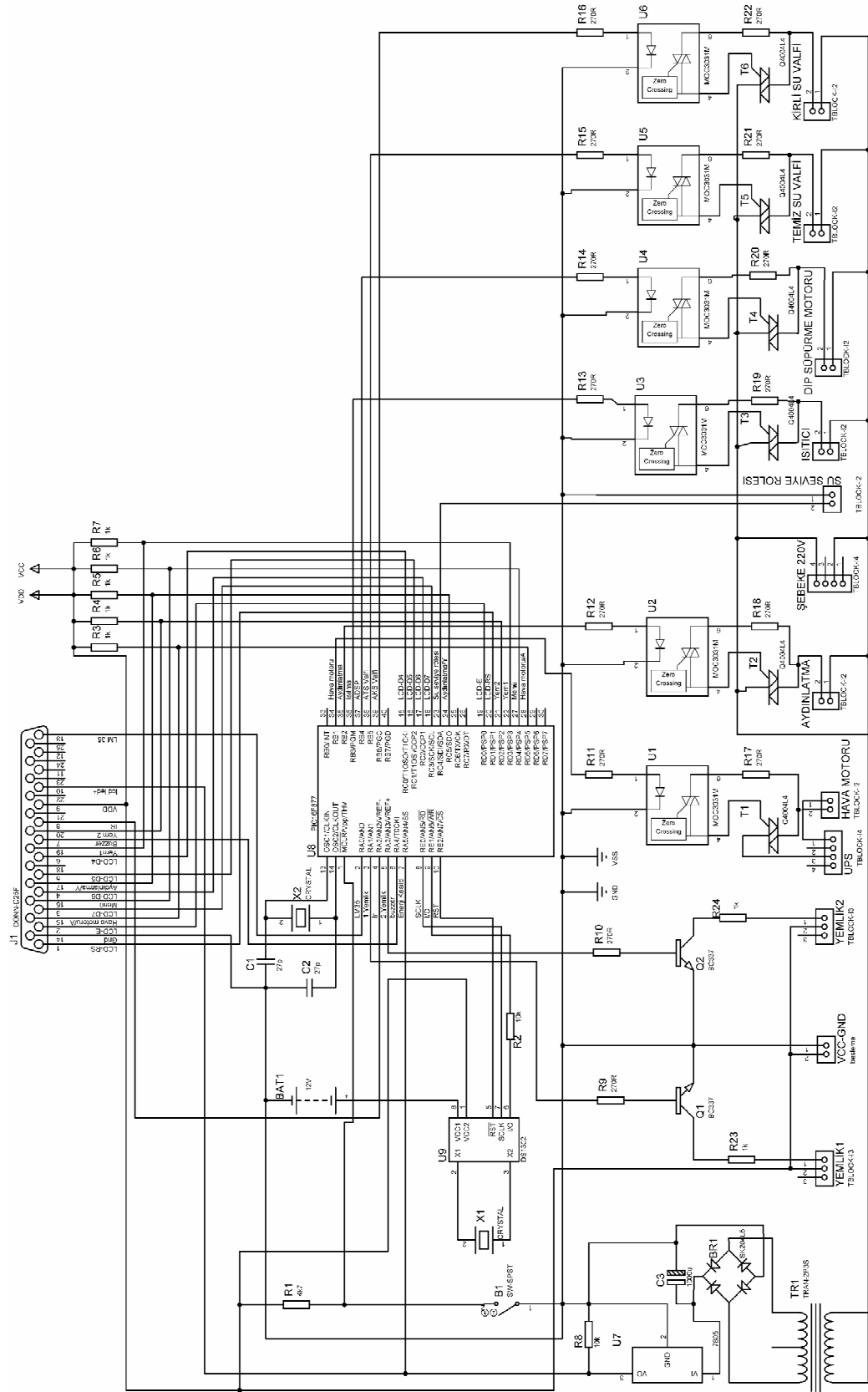
Akvaryumdaki bütün işlemler ve ayarlar bu kontrol paneli üzerinden yapılabilmektedir. Menü işlemlerinin dışındaki tek tuş işlemleri uzaktan kumanda ile de yapılabilmektedir.

Uzaktan kumanda sinyalleri IR tarafından alınıp mikrodenetleyiciye gönderilmektedir. Led ise IR alıcının sinyalleri aldığını yanarak göstermektedir. LDR ise hava karardığında LCD ekranın arka panel lambalarının ortamdaki ışığa göre yanmasını sağlayarak hoş bir görüntü oluşturur. Isı sensörü ise sıcaklığını algılayarak mikrodenetleyiciye gönderir.

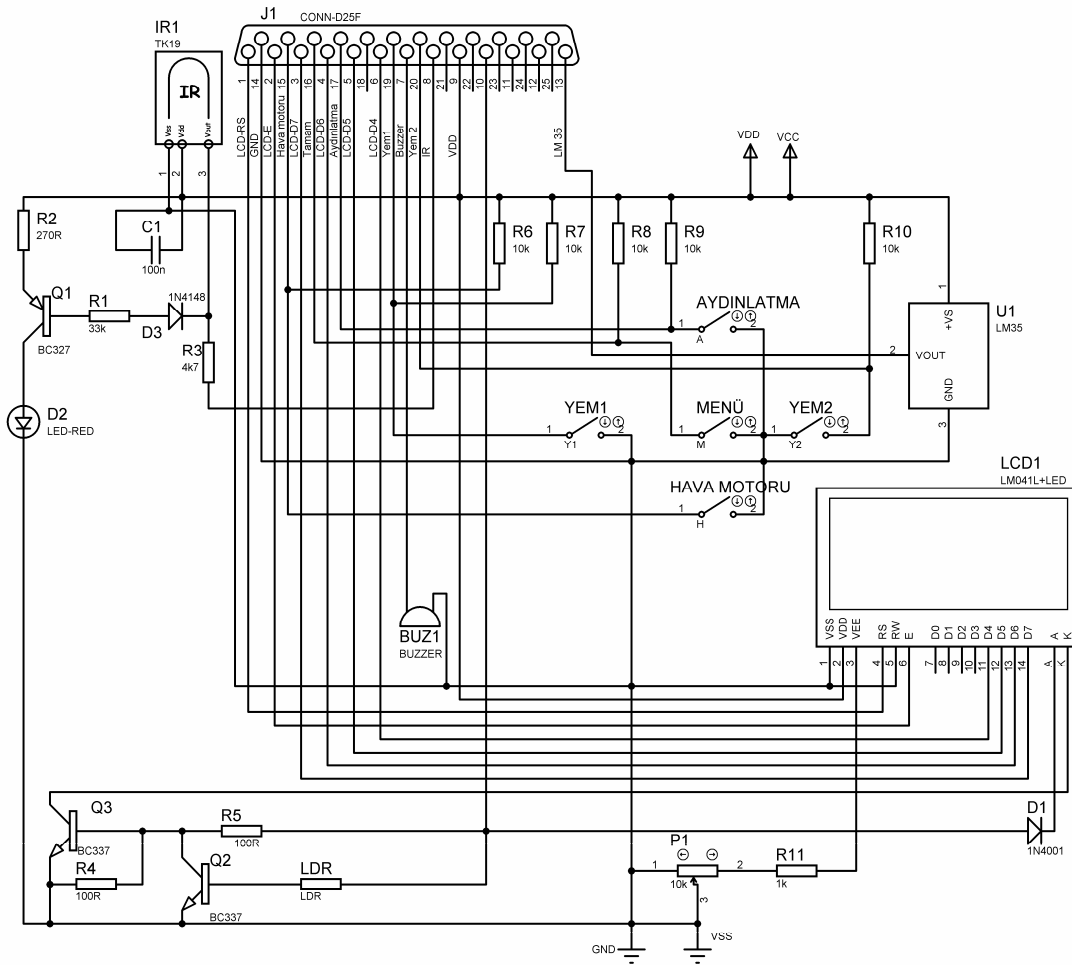


Şekil 4.4. Göstergeli kontrol paneli

PIC 16F877 mikrodenetleyicili anakart devresi ve göstergeli kontrol paneli devresinin açık şemaları şekil 4.5 ve şekil 4.6'da görülmektedir.



Şekil 4.5. PIC 16F877 mikrodeneleyicili anakart devresi açık şeması



Şekil 4.6. Göstergeli kontrol paneli devresi açık şeması

PIC 16F877 mikrodenetleyicili anakart devresinin eleman listesi tablo 4.1’de ve göstergeli kontrol paneli devresinin eleman listesi tablo 4.2’de verilmiştir. Kullanıcı işlemlerini göstergeli kontrol panelindeki 4x16 LCD (Liquid Crystal Display) ve 5 adet tuş yardımıyla kolaylıkla yapabilmektedir. Bu tuşlar, yemlik 1, yemlik 2, aydınlatma, hava motoru ve menüdür. Tasarımda iki adet yemlik vardır. Bu yemliklerden bir tanesi yavru balıklar için tasarlanmıştır. Yemleme; uzaktan kumandadan, göstergeli kontrol panelinden ve önceden programlanan zamanda yapılabilmektedir. Aydınlatma ve hava motoru, hem uzaktan kumandadan hem de göstergeli kontrol panelinden kontrol edilebilmektedir. Su değişimi haftada bir kez yapılmaktadır. Bunun zamanı önceden göstergeli kontrol panelinden ayarlanmaktadır.



Tablo 4.1. Anakart devresinin eleman listesi

ELEMANNIN	
İSMİ	DEĞERİ
R1	4,7k $\Omega$
R2, R8	10k $\Omega$
R3-R7, R23, R24	1k $\Omega$
R9-R22	270 $\Omega$
C1, C2	27pF
C3	1000 $\mu$ F
U1-U6	MOC3031M
U7	7805
U8	PIC16F877
Aydınlatma	TBLOCK-I2
Dip süpürme motoru	
Hava motoru	
Isıtıcı	
Kirli su valfi	
Temiz su valfi	
Su seviye rölesi	
B1	
BAT1	3,6V
BR1	SK204L5
J1	CONN-D25F
T1-T6	Q4004L4
TR1	TRAN-2P3S
UPS, Şebeke 220V	TBLOCK-I4
VCC-GND	Besleme
X1	32768Hz CRYSTAL
X2	4MHz CRYSTAL
Yemlik1, Yemlik2	TBLOCK-I3

Tablo 4.2. Göstergeli kontrol paneli devresinin eleman listesi

ELEMANNIN	
İSMİ	DEĞERİ
R1	33k $\Omega$
R2	270 $\Omega$
R3	4,7k $\Omega$
R4, R5	100 $\Omega$
R6-R10	10k $\Omega$
R11	1k $\Omega$
C1	100nF
U1	LM35
Q1	BC327
Q2, Q3	BC337
D1	1N4001
D2	LED-RED
D3	1N4148
LDR	LDR
Buz1	BUZZER
P1	10k $\Omega$
IR1	TK19
J1	CONN-D25F
LCD1	LM041L+LED
Hava motoru	H
Menü	M
Aydınlatma	A
Yem1	Y1
Yem2	Y2

#### 4.1. PIC 16F877 Mikrodenetleyici

PIC 16F877 Mikrodenetleyici bölüm 3'te ayrıntılı olarak incelenmiştir.

#### 4.2. 4x16 Likit Kristalli Ekran

Likit kristalli ekran; günlük yaşamda cep telefonlarında, beyaz eşyalarda, güvenlik sistemlerinde ve birçok elektronik sistemde kullanılan ve kendi karakter hafızası bulunan bir birimdir. 4X16; 4 satır ve 16 sütundan oluşan bir ekran anlamına gelmektedir. Likit kristalli ekran 16 adet bağlantıya sahiptir. Bu bağlantılar aşağıda açıklanmıştır.

1. ve 2. bacaklar güç kaynağı hatları yani Vss ve Vdd' dir. Vdd bacağına pozitif gerilime, Vss' nin de 0 V' a veya toprağa bağlanması gerekir. Likit kristalli ekran birimleri veri kitapçıklarının çoğunda kaynak / besleme gerilimi 5V gösterilmesine rağmen, 6V ve 4.5V'luk beslemelerde de oldukça iyi çalışmaktadır. Hatta bazı ekran birimlerinde besleme gerilimi 3V'a kadar düşmektedir. Bu nedenle ekran birimleri etkin ve ekonomik olarak pil / batarya ile beslemek de mümkündür.

3. bacak yani Vss, ekranın parlaklığını ayarlamaya yarayan bir kontrol ucudur. Bu bacak değişken bir gerilim kaynağına veya besleme hatları arasına bağlanan bir ayarlı direncin orta ucuna bağlanarak bu ayar yapılabilmektedir. Ancak bazı likit kristalli ekran birimlerinin -7 V'a varan gerilimlere ihtiyaç duyduğu da göz önüne alınırsa, en basit olarak bu bacağın 0V'a bağlanması en uygundur. Bu devrede 10k $\Omega$ 'luk P<sub>1</sub> potansiyometresi ile ekranın parlaklığı ayarlanmaktadır.

4., 5. ve 6. bacaklar komut kontrol bitleri olarak isimlendirilebilirler. Bunlardan 4. bacak yani RS yazmaç seçme bitidir ve bu komut kontrol bacaklarının ilkinin oluşturur. Bu hat düşük (Lojik 0) yapıldığı durumda ekrana aktarılan veri bitleri komut olarak algılanır ve gerekli işlem yerine getirilir. Bu durumda ekrandan okunan veri bitleri ise, ekranın durumu hakkında bilgi verir. Bu hattın yüksek (Lojik 1) yapılması ile de, birime karakter veri transferi veya alımı yapılacağı anlaşılır. Kısa ve basitçe özetlemek gerekirse, ekranda bir karakter yazmak veya ekrandan bir karakter

okumak için RS hattı yüksek, ekrana bir komut yollamak veya ekranın durumu hakkında bilgi almak istersek RS hattını düşük yapmamız gerekmektedir. 5. bacak yani R/W hattı, kısaca oku / yaz anlamına gelmektedir. Eğer ekrana karakter veri transferi yapılacaksa veya bir komut yollanacaksa düşük, karakter veri alımı yapılacaksa veya yazmaçlardan durum bilgisi okunacaksa yüksek yapılır. 6. bacak yani E ise komut kontrol bitlerinin sonuncusunu oluşturur ve yetki biti olarak isimlendirilebilir. Bu giriş, birim ve veri hatları arasında, komutların veya karakter verilerinin, gerçek anlamda aktarımını başlatmak için kullanılır. Ekrana yazılırken, veri aktarımı sadece bu işaretin düşen kenarında gerçekleşir. Bununla birlikte, ekrandan okuma yapılırken, veri yükselen kenar hemen kısa bir süre sonra hazır olur ve işaret tekrar düşüncüye kadar hatta kalır.

7. ile 14. bacaklar arasındaki uçlar sekiz adet veri hattıdır. Veri ekrana, ya 8 bit'lik tek bir byte olarak ya da, içi 4 bit'lik nibble'lar olarak aktarılır veya ekrandan okunur. Bu ikinci durumda, sadece üst dört veri hattı (D4' den D7' ye ) kullanılır. Bu 4 bit modu, bir mikrodenetleyici kullanıldığında, daha az giriş / çıkış hattına gerek olduğundan kullanışlıdır. Bu devrede D4, D5, D6 ve D7 uçları kullanılmıştır.

15. ve 16. bacaklar ise aydınlatma girişleridir. 15. bacak +5 Volt gerilime, 16. bacak ise toprak hattına bağlandığında ekran ışıklı hale gelmektedir.

LCD ekranında gün, ay, yıl olarak tarih bilgisi, haftanın hangi günü olduğu bilgisi, saat bilgisi, aydınlatmanın ve hava motorunun çalışıp çalışmadığı (+ ve - olarak) bilgisi, elektrik enerjisinin olup olmadığı bilgisi (enerji varken sistem normal, enerji yokken enerji kesildi yazmaktadır) ve ortamın sıcaklık bilgisi bulunmaktadır. Ayrıca ekranda yem atılırken “yem atıyorum”, su değişimlerinde “su bosaltıyorum” ve “su dolduruyorum” yazmaktadır.

### 4.3. Tuşlar

Kullanıcı ile sistem arasındaki etkileşim beş adet tuş ile sağlanmaktadır. Bu tuşlara verilen isimler ve görevleri aşağıda anlatılmıştır.

#### **4.3.1. Menü (M)**

Bu tuş ile tarih, saat, gün, yem atış ayarına, su değişim zamanının ayarlarına giriş yapılmaktadır. Ayarlardan ana menüye hızlı çıkış için de kullanılır. Bu tuş diğer 4 tuşun ortasında yer almaktadır.

#### **4.3.2. Aydınlatma (A)**

Aydınlatma kapalı iken bu tuşa basıldığında aydınlatma açılır. Aydınlatma açık iken bu tuşa basıldığında aydınlatma kapanır. Aydınlatma tuşu diğer 4 tuşun yukarısında yer almaktadır. Bu tuş tarih, saat, yem atış zamanının belirlenmesinde ve su değişim zamanının belirlenmesinde rakamları arttırmak için kullanılır.

#### **4.3.3. Hava motoru (H)**

Hava motoru çalışmadığı zaman bu tuşa basıldığında hava motoru çalışmaya başlar. Hava motoru çalışır iken bu tuşa basıldığında hava motoru durur. Hava motoru tuşu diğer 4 tuşun aşağısında yer almaktadır. Bu tuş tarih, saat, yem atış zamanının belirlenmesinde ve su değişim zamanının belirlenmesinde rakamları azaltmak için kullanılır.

#### **4.3.4. Yemlik 1 (Y<sub>1</sub>)**

Yemleme işlemi için yavru balıklar düşünülerek sistemde 2 adet yemlik kullanılmıştır. 1. yemlik çalıştırılmak istenildiğinde yemlik 1 tuşuna basılır ve yemlik 1 çalışır. Yemlik 1 tuşu diğer 4 tuşun solunda yer almaktadır. Bu tuş tarih, saat, yem atış zamanının belirlenmesinde ve su değişim zamanının belirlenmesinde sola doğru gitmek için kullanılır.

#### **4.3.5. Yemlik 2 (Y<sub>2</sub>)**

2. yemlik çalıştırılmak istenildiğinde yemlik 2 tuşuna basılır ve yemlik 2 çalışır. Yemlik 2 tuşu diğer 4 tuşun sağında yer almaktadır. Bu tuş tarih, saat, yem atış

zamanının belirlenmesinde ve su deęişim zamanının belirlenmesinde saęa doęru gitmek için kullanılır.

#### 4.4. Giriş Devreleri

##### 4.4.1. Enerji kontrol

Elektrik enerjisi varken anakart ve göstergeli kontrol paneli devreleri enerjilerini şekil 4.7’de görülen güç kaynağından almaktadır. Bu durumda LCD’de “sistem normal” yazmaktadır.



Şekil.4.7. Güç kaynağı

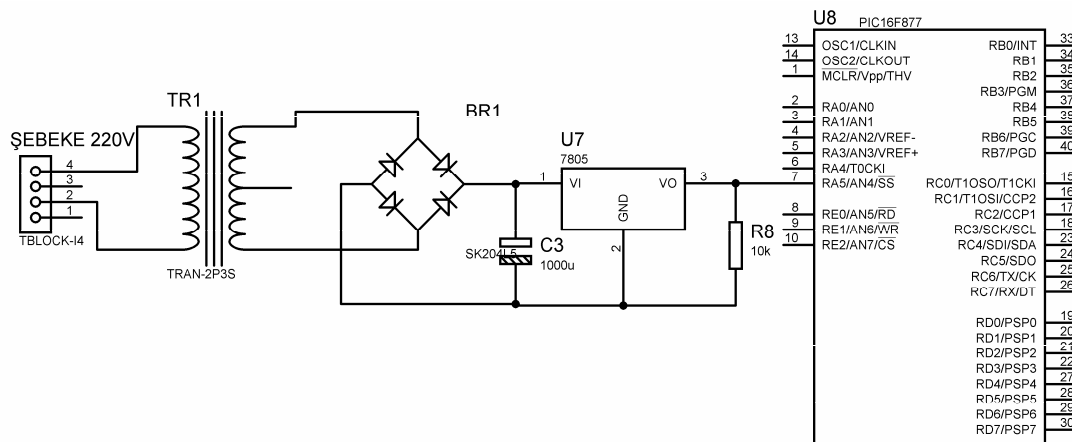
Elektrik enerjisi kesildiğinde yemleme ve havalandırma işlemlerinin devam edebilmesi için şekil 4.8’de görülen kesintisiz güç kaynağı (UPS) kullanıldı. Elektrik enerjisi kesildiğinde kesintisiz güç kaynağı devreye girmektedir ve LCD ekranda “enerji kesildi” yazmaktadır. Kesintisiz güç kaynağı devreye girdiğinde beş saat ara ile beş dakika hava motoru çalışmakta ve sadece programlanan zamanlarda yemleme işlemleri yapılmaktadır. Bu durumda göstergeli kontrol panelindeki yemlik1, yemlik2, hava motoru ve aydınlatma tuşlarına basıldığında yemleme, havalandırma

ve aydınlatma yapılamamaktadır. Sadece menü fonksiyonu çalışmaktadır ve bu tuşu basarak tarih, saat, yemleme saati ve su değişim saati ayarları yapılabilmektedir.



Şekil 4.8. Kesintisiz güç kaynağı (UPS)

Elektrik enerjisinin olup olmadığı şekil 4.9’da görülen enerji kontrol devresi ile belirlenmektedir. Şebekede 220V olduğu zaman enerji kontrol devresindeki transformatör, köprü diyot, kondansatör iletime geçer ve 7805’in çıkışında 5V olur. PIC 16F877 mikrodenetleyici bu 5V’u (lojik 1) algılar ve devre “sistem normal” modunda çalışır. Elektrik enerjisi kesildiğinde yani şebekede 220V yokken transformatörün sekonder uçlarında gerilim düşümü olmaz. Dolayısıyla köprü diyot yalıtkan olur ve 7805’in çıkışı 0V olur. PIC 16F877 mikrodenetleyici bu 0V’u (lojik 0) algılar ve devre “enerji kesildi” modunda çalışır.



Şekil 4.9. Enerji kontrol devresi

#### 4.4.2. Su seviye rölesi

Su deęişimlerinde suyun hangi seviyeye kadar doldurulacağı ve boşaltılacağı şekil 4.10'daki su seviye rölesi yardımıyla yapılmaktadır.

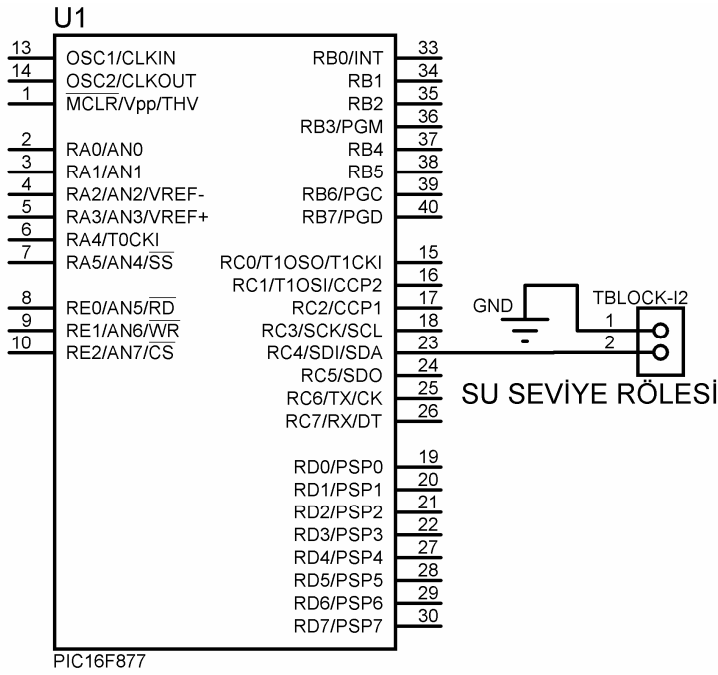


Şekil 4.10. Su seviye rölesi

Su seviye rölesindeki Ü, A ve T'nin anlamları sırasıyla üst, alt ve topraktır. (T) Toprak ucu akvaryumun dibine yerleştirilir. Su deęişimlerinde su nereye kadar boşaltılacaksa A (alt) ucu o noktaya yerleştirilir. Su deęişimlerinde su nereye kadar doldurulacaksa Ü (üst) ucu o noktaya yerleştirilir. Su deęişimlerinde deęişim miktarı %20 tavsiye edilmektedir. Akvaryumun ölçülerine dikkat ederek üst ve alt ucundaki mesafe ayarlanır. L ve N'nin anlamları faz ne nötrdür. L ve N ucundaki fiş prize faz ve nötre dikkat ederek takılır. 1 ve 2 numaralı uçların bağlantısı şekil 4.11'de görülmektedir. Su deęişim zamanında su, üst seviyede, üst ve alt seviye arasında ve de alt seviyenin altında olabilir. Eğer su üst seviyede ise su deęişim zamanında su, önce alt seviyeye kadar şekil 4.12'deki su boşaltma motoru ile boşatılır sonra üst seviyeye kadar doldurulur. Su deęişim zamanında eęer su dięer iki seviyeden birinde ise su, önce üst seviyeye kadar doldurulur, alt seviyeye kadar boşatılır ve sonra



tekrar üst seviyeye kadar doldurulur. Su değişimi yapılmadığı zamanlarda ve su boşaltılırken su seviye rölesindeki röle kontakları açık ve röle ledi söndür. Su boşaltımında su alt seviyeye gelince röle kontakları kapanır, röle ledi yanar ve su doldurulmaya başlanır. Su üst seviyeye ulaşınca röle kontakları açılır ve röle ledi söner. Böylece su değişimi tamamlanmış olur.



Şekil 4.11. Su seviye rölesinin devre şeması

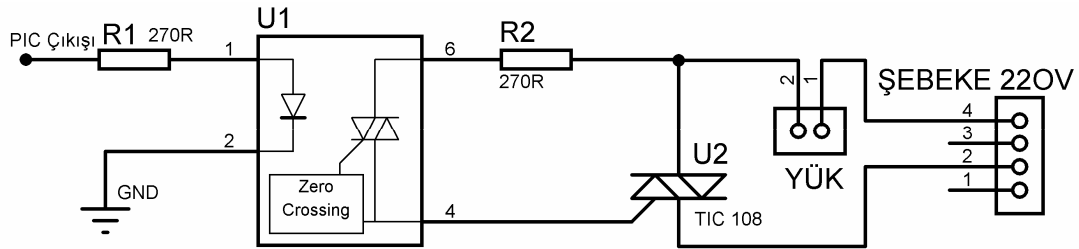


Şekil 4.12. Su boşaltma motoru

## 4.5. Çıkış Devreleri

### 4.5.1. 220V ile çalışan çıkış devreleri

220V ile çalışan çıkış devreleri hava motoru, aydınlatma, ısıtıcı, dip süpürme motoru, kirli su valfi ve temiz su valfidir. Bu devrelerin açık şemaları şekil 4.13'te görülmektedir. Kullanılan optokuplör ile şebeke ile lojik devre arasında yalıtım yapmaktadır. PIC 16F877 mikrodenetleyici çıkışı lojik 1 olduğunda optokuplör ilettime geçer. Triyak tetiklenerek ilettime geçince çıkışında yük çalışır.



Şekil 4.13. 220V ile çalışan çıkış devrelerin açık şemaları

#### 4.5.1.1. Hava motoru

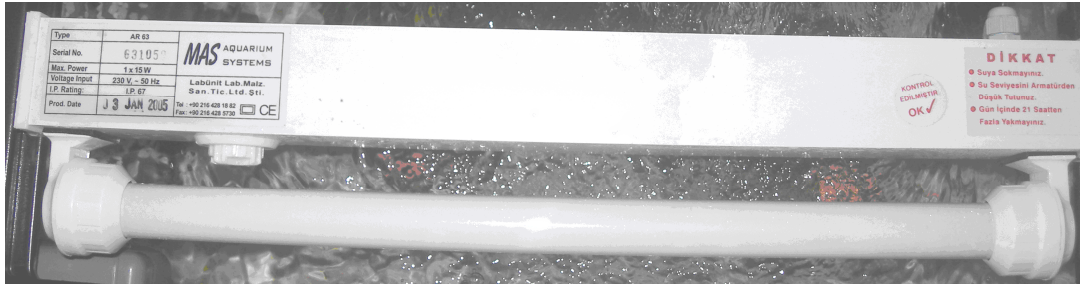
Akvaryum otomasyonu tasarımında kullanılan hava motoru şekil 4.14'de görülmektedir. Hava motoru göstergeli kontrol panelindeki H tuşu ile çalıştırılıp durdurulabilmektedir. Aynı zamanda uzaktan kumanda ile de hava motoru kontrol edilebilmektedir.



Şekil 4.14. Hava motoru

#### 4.5.1.2. Aydınlatma

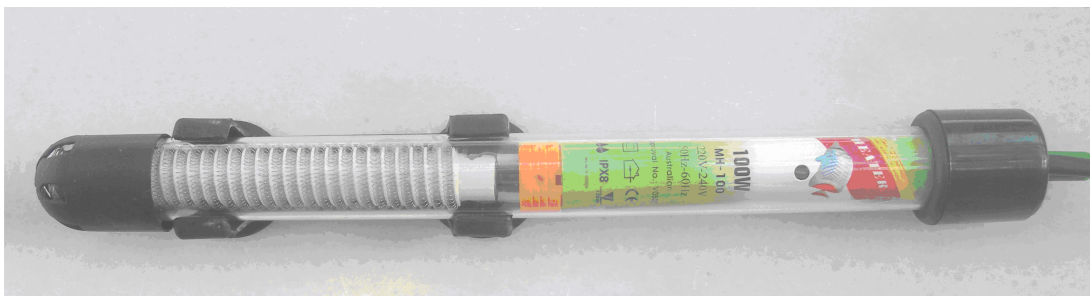
Şekil.15’de akvaryum otomasyonu tasarımında aydınlatmada kullanılan flüoresan görülmektedir. Aydınlatma göstergeli kontrol panelindeki A tuşu ile çalıştırılıp durdurulabilmektedir. Aynı zamanda uzaktan kumanda ile de aydınlatma kontrol edilebilmektedir.



Şekil 4.15. Flüoresan

#### 4.5.1.3. Isıtıcı

Akvaryum otomasyonu tasarımında kullanılan ısıtıcı şekil 4.16’da görülmektedir. Isıtıcı akvaryum suyunun sıcaklığını dengede tutmak için kullanılmaktadır. Su sıcaklığı 25°C seviyesinde olması istenir. Tasarımda kullanılan ısıtıcı 100W’lıktır.



Şekil 4.16. Isıtıcı

#### 4.5.1.4. Dip süpürme motoru

Dip süpürme motoru akvaryumun sol alt köşesine yerleştirilmiştir. Su boşaltma motoru ve dış filtrenin kirli suyu çekme ucu akvaryumun sağ alt köşesine yerleştirilmiştir(Şekil 1.1). Su boşaltma zamanı gelince su boşaltma motoru

çalışmadan yaklaşık 10 saniye önce şekil 4.17’de görülen dip süpürme motoru çalışmaya başlar ve su boşaltma süresince çalışmaya devam eder. Akvaryumda bulunan pislikler dip süpürme motoru ile sol taraftan su boşaltma motoru ve dış filtrenin kirli suyu çekme ucunun bulunduğu sağ tarafa doğru süpürülürler. Böylece akvaryumun pisliklerden temizlenmesi sağlanır.



Şekil 4.17. Dip süpürme motoru

#### 4.5.1.5. Kirli su valfi

Şekil 4.18’de akvaryum otomasyonu tasarımında kullanılan kirli su valfi görülmektedir. Su boşaltımı süresince kirli su valfi açılır ve su boşaltımı gerçekleştirilir.



Şekil 4.18. Kirli su valfi

#### 4.5.1.6. Temiz su valfi

Akvaryum otomasyonu tasarımında kullanılan temiz su valfi şekil 4.19'da görülmektedir. Su dolumu sırasında temiz su valfi açılır ve su dolumu gerçekleştirilir.

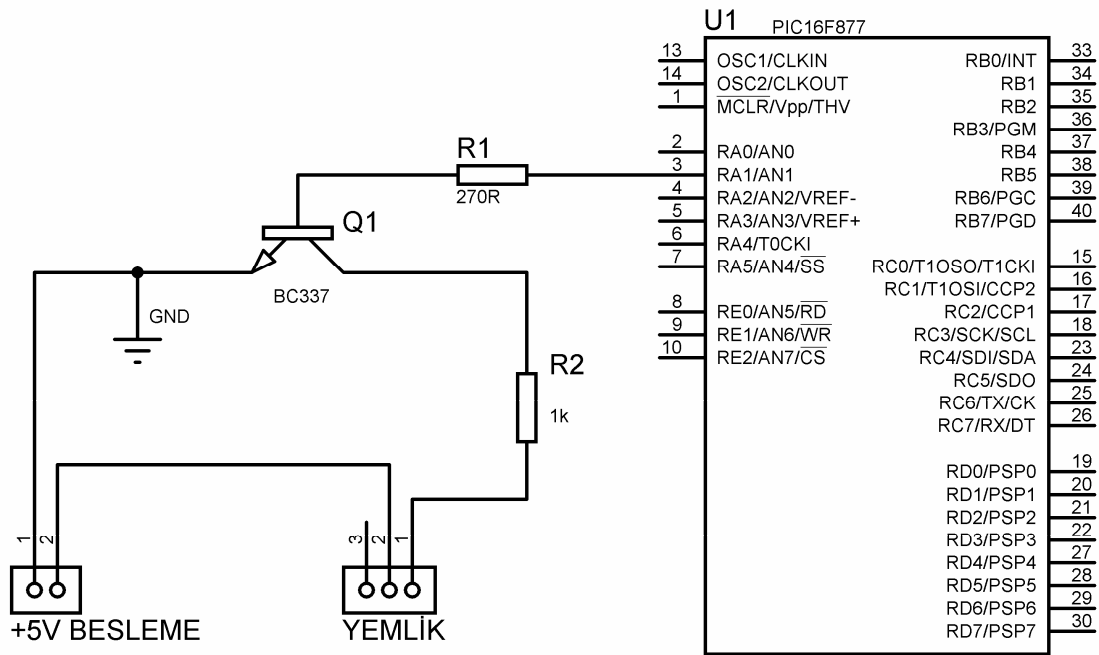


Şekil 4.19. Temiz su valfi

#### 4.5.2. 5V ile çalışan çıkış devreleri

5V ile çalışan çıkış devreleri yemlik 1 ve yemlik 2 devreleridir. Bu devrelerin açık şemaları şekil 4.20'de görülmektedir. Göstergeli kontrol panelinde  $Y_1$  veya  $Y_2$  tuşlarına basıldığında, yemleme zamanı geldiğinde veya uzaktan kumandadan yemleme tuşlarına basıldığında PIC 16F877 mikrodenetleyici çıkışı lojik 1 olduğundan transistör iletime geçer. Böylece anahtarlama elemanı olarak kullanılan transistör iletime geçince yemlik 1 veya yemlik 2 çalışır. PIC 16F877 mikrodenetleyicinin 3 numaralı bacağı yemlik 1, 5 numaralı bacağı yemlik 2 için kullanılmıştır. Yemlik 1 ve yemlik 2 devreleri eşdeğerdir. 2 adet yemlik kullanılmasının sebebi, yemliklerden birinin yavru balıkların yemlenmesinde kullanılmak istenmesidir. Tasarımda kullanılan yemlikler şekil 4.21'de görülmektedir.





Şekil 4.20. 5V ile çalışan çıkış devrelerin açık şemaları



Şekil 4.21. Yemlik

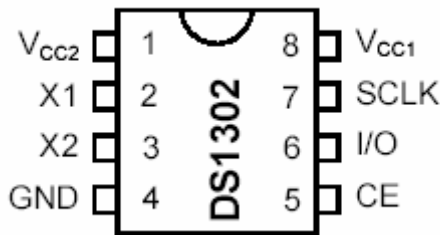
## 4.6. Sistemdeki Diğer Devreler ve Elemanlar

### 4.6.1. DS1302 saat ve takvim devresi

DS1302, Dallas yarıiletken firması tarafından üretilmiş olan bir gerçek zamanlı saat ve takvim devresidir. 8 bacaklı bir devre olan DS1302, 2100 yılına kadar çalışma garantisine sahiptir ve bacak sayısını ekonomik olarak kullanabilmek için seri iletişim yeteneğine sahiptir. 2 Volt ile 5,5 Volt arası besleme gerilimine ihtiyaç duyan devre; -40 °C ile +85 °C arası çalışabilme özelliğine sahiptir.

Devre besleme gerilimi kesildiğinde saat ve takvim bilgilerini saklayabilmek için yapısında rasgele erişimli bellek ( RAM ) bulundurmaktadır. Bu bellek sürekli çalışmak için tasarlanmıştır ve Vcc1 besleme gerilimine ihtiyaç duymaktadır. Ana besleme gerilimi yani Vcc2 kesintiye uğradığında, Vcc1 bacağına bağlanmış olan pil, saati ve takvim bilgilerini içinde barındıran belleği sürekli aktif tutacaktır ve dolayısıyla bilgi kaybı olmadan saat ve takvim çalışmaya devam edecektir. Vcc2 bacağına bağlanacak olan en az 1,5 Volt'luk pilin, yeniden şarj edilebilir pil olması tavsiye edilmektedir. Çünkü DS1302 devresi pil şarj etme düzeneğine sahiptir. Devrenin katalog bilgileri incelendiğinde, şarj etme düzeneğinin nasıl çalıştığı kolayca anlaşılacaktır. Gerekli yazılımsal ayarlamalar yapılmadan, şarj edilebilir pil kullanmanın bir anlamı yoktur. Bu ayarlamalar devrenin katalog bilgilerinde bulunmaktadır.

X1 ve X2 bacakları osilatör bacaklarıdır. 32.768 KHz frekansında çalışan bir osilatör, saat ve takvim tüm devresinin çalışması için gereklidir. Şekil 4.22'de DS1302 tüm devresi görülmektedir.



Şekil 4.22. DS1302 Tüm devresi

#### 4.6.2. Uzaktan kumanda ve IR (Uzaktan kumanda alıcısı – TK19)

Sistemde kullanılan uzaktan kumanda ve uzaktan kumandanın tuş takımı Şekil 4.23'te görülmektedir. Yem atma, havalandırma, aydınlatma, işlemleri uzaktan kumanda ile yapılabilmektedir. Göstergeli kontrol panelindeki gibi sol tuş  $Y_1$ , sağ tuş  $Y_2$ , yukarı tuş aydınlatma ve aşağı tuş hava motorudur.



Şekil4.23. Uzaktan kumanda (a), Uzaktan kumandanın tuş takımı (b)

Uzaktan kumanda sinyalleri IR tarafından alınıp mikrodenetleyiciye gönderilmektedir. Kullanılan IR elmanı TK19'dur.(Şekil 4.24) TK19'da 1 numara GND, 2 numara  $V_{DD}$  ve 3 numara  $V_{out}$  ucudur. TK19 38 KHz frekansında çalışmaktadır.[13]



Şekil 4.24. TK19



## BÖLÜM 5. SONUÇ VE ÖNERİLER

Bu çalışmada bir akvaryumda bulunan balıkların aydınlatma, havalandırma, yemleme, filtreleme, ısıtıcı ve su değişimi ihtiyaçlarını karşılayacak akvaryum otomasyonu tasarlanmış ve yapılmıştır. Öncelikle sayılan bu ihtiyaçları giderecek PIC 16F877 mikrodenetleyicili devre tasarlanmış ve program yazılmıştır.

Akvaryum otomasyonu ile biri yetişkin biri yavru balıklar için hazırlanan iki yemlikten günün istenilen saatlerinde yemleme işlemi ve akvaryumun haftalık su değişimini otomatik olarak yapılabilmektedir. Tasarımda bulunan dip süpürme moturu, su değişimlerinde akvaryumdaki pislikleri su boşaltma motoruna doğru süpürerek pisliklerin atılmasını ve böylece akvaryumun daha uzun süre temiz kalmasını sağlamaktadır.

Ayrıca tasarımda bulunan göstergeli kontrol paneli sayesinde, sistem için gerekli saat ve tarih ayarları, yemliklerin günün hangi saatinde yem atacakları ayarı ve su değişiminin gün ve saat ayarı yapılabilmektedir. Yem atma, havalandırma ve aydınlatma işlemleri göstergeli kontrol panelinden tek tuşla yapılabilmektedir. Göstergeli kontrol paneli devresinin plaketi, üstü 3 mm kalınlığında fleksiglass malzeme ile kaplanarak hem estetik bir görüntü sağlanmış hem de plaketin mukavemeti arttırılmıştır.

Yem atma, havalandırma ve aydınlatma işlemleri uzaktan kumanda ile de yapılabilmektedir. Su değiştirme mekanizması direkt olarak temiz ve pis su şebekesine bağlanarak, bu sayede sisteme müdahale edilmeden daha uzun süreli bir otomasyon sağlanmıştır. Enerji kesilmelerinde, kesintisiz güç kaynağı ile sistem desteklenip havalandırma ve yemleme yapılabilmektedir.

Sisteme RS232 protokolünü destekleyen MAX232 entegresi eklenirse bu bağlantı

noktası üzerinden akvaryumun bilgisayar ile kontrolü sağlanabilecektir. Bilgisayar üzerindeki TV kartı ile veya web kamerası ile başka odadan veya internet üzerinden kamera ile akvaryum seyredilebilecektir. Sisteme eklenecek ikinci bir elektronik devre ile uyarılar, basit sesler yerine gerçek sesler ile duyulabilecektir. Bu sesler istenilirse bizim sesimiz, istenilirse çocuklarımızın sesi istenilirse de hoş bir melodi olabilir.

Sisteme eklenebilecek ikinci bir kesintisiz güç kaynağı ile elektrik kesintilerinde ısıtıcı ve filtrenin çalışması sağlanabilir. Ayrıca hava motoru ve aydınlatmanın günün ayarlanan saatlerinde çalışması ve durması ayarlanabilir.

Sonuç olarak yoğun iş yükü olan veya uzun süreli gezilere çıkan insanların evlerinde bulunan akvaryumlarının, önceden hazırlanmış programa göre çalışması sağlanmıştır.

## KAYNAKLAR

- [1] <http://www.bilyap.com.tr/freshwater/tatlisu.php>
- [2] <http://www.aadf.net/projeler.htm>
- [3] [http://www.akvaryumkulubu.org/index.php?belge\\_id=23](http://www.akvaryumkulubu.org/index.php?belge_id=23).
- [4] [http://www.akvaryumkulubu.org/index.php?belge\\_id=24](http://www.akvaryumkulubu.org/index.php?belge_id=24).
- [5] [http://www.akvaryumkulubu.org/index.php?belge\\_id=543](http://www.akvaryumkulubu.org/index.php?belge_id=543).
- [6] [http://www.akvaryumkulubu.org/index.php?belge\\_id=27](http://www.akvaryumkulubu.org/index.php?belge_id=27).
- [7] <http://www.bilyap.com.tr/freshwater/donanim.php#B3>.
- [8] Sera rehber, "Akvaryumumu nasıl kurarım" sayfa 18-19.
- [9] [http://www.akvaryumkulubu.org/index.php?belge\\_id=28](http://www.akvaryumkulubu.org/index.php?belge_id=28).
- [10] İstanbul Teknik Üniversitesi, Elektrik – Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü, Elektronik Mühendisliği Programı, Oğuzhan Kızılbey, Bitirme Ödevi, Deniz Suyu Termometresi, Mayıs 2005
- [11] [http://robot.ee.hacettepe.edu.tr/forum/files/18\\_3\\_Pic%2016F877%20ve%20%C3%96zellikleri.pdf](http://robot.ee.hacettepe.edu.tr/forum/files/18_3_Pic%2016F877%20ve%20%C3%96zellikleri.pdf)
- [12] Akıllı sayma rölesi tasarımı ve PIC16F877 Mikrodenetleyicisi ile gerçekleştirilmesi. İsmail Yavuz, Gökhan Gelen ve Murat Uzam. Elektrik Elektronik Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Niğde Üniversitesi, 51200 NİĞDE
- [13] [http://www.biltek.tubitak.gov.tr/gelisim/elektronik/dosyalar/18/IR\\_kumanda.pdf](http://www.biltek.tubitak.gov.tr/gelisim/elektronik/dosyalar/18/IR_kumanda.pdf)

## EKLER

### EK A. C DOSYALARI

#### A.1. Ana Program

```
/******  
* Akvaryum Otomasyonu Tasarımı ve Yapımı Mayıs 2007*  
* Faruk AKYILDIZ *  
* Tez Danışmanı: Yrd. Doç. Dr. Ali Fuat BOZ *  
*****/  
  
// Eklenen kütüphaneler  
#include <pic.h>  
#include "delay.h"  
#include "lcd.h"  
#include "Akvaryum.h"  
#include "LCDGosterge.h"  
#include "Ses.h"  
#include "DS1302.h"  
#include "UKumanda.h"  
#include "DKontrol.h"  
#include "ETazele.h"  
  
// Alt Programlar *****/  
// Ekran yenileme ve durum kontrolleri -----  
Yenile(void)  
{  
    Skont=clkreadbyte(0); Skont&=0x0F;  
    if((Skont==0)||((Skont==5) // : yaz  
        {
```

```

        lcd_goto(2); lcd_putchar(':'); lcd_goto(96);
    }
    if((Skont==2)||((Skont==7))           // : sil
        {
            lcd_goto(2); lcd_putchar(' '); lcd_goto(96);
        }
    if(Skont==9)
        {
            LCD(2,64);           // Isı deęişimini yenile
//            LCD(18,80);        // Isıyı kontrol et
            LCD(8,80);           // Sistem durumunu Yenile
            DKon(0);             // Enerji Kesildi mi ?
        }
    if(clkreadbyte(1)!=Zkont)
        {
            Zkont=clkreadbyte(1);
            LCD(0,0);           // Saat deęişimini Yenile
            LCD(1,8);           // Tarih Deęişimini Yenile
//            LCD(2,64);        // Isı deęişimini yenile
//            LCD(18,80);       // Isıyı kontrol et
            LCD(17,71);         // (17,23) Haftanın Gününü Yazdır
            for(Say=0;Say<13;Say=Say+4)
                {
                    if((eeprom_read(10+Say)==ZamanO(2))&&(eeprom_read(11+Say)==ZamanB(2))&
&(eeprom_read(12+Say)==ZamanO(1))&&(eeprom_read(13+Say)==ZamanB(1)))
                        {
                            if(('0'==ZamanO(2))&&('0'==ZamanB(2))&&('0'==ZamanO(1))&&('0'==ZamanB(1
                            ))) goto Yn;
                                switch(Say)
                                    {
                                        case 0 :      DKon(1);break           // Yem1 at
                                        case 4 :      DKon(2);break;           // Yem2 at
                                        case 12: if((eeprom_read(26)==clkreadbyte(5))

```

```

        &&(eeprom_read(0)==0)                // Su deđiřtir
    {
        DKon(5);break;
    }
    }
}

Yn : asm("NOP");
}
// Alt Memü Tuř Kontrolleri -----
MKay(unsigned char HMenu, unsigned char Kay)
{
    // 0,15,31,95,0 HMenu>2 ise
    unsigned char MKOYer[5]={0,15,79,31,95};
    if(HMenu>2)
    {
        MKOYer[2]=31;
        MKOYer[3]=95;
    }
    lcd_goto(MKOYer[Kay]);
}
// G¼n yazdırma -----
GYaz(unsigned char GYPoz,unsigned char GYYaz)
{
    lcd_goto(GYPoz);
    if (Tkont==1) for(Say=0;Say<3;Say++) lcd_putch(gunler[GYYaz][Say]);
// else for(Say=0;Say<4;Say++) lcd_putch(ilac[GYYaz][Say]);
    lcd_goto(GYPoz); Tkont=0;
}
// Eepromdan oku Ekran yaz (0!=) - Ekrandan Oku Eeprom yaz ( 0== )-----
ZYukle(unsigned char ZYPoz,unsigned char ZY)
{
    unsigned char YSay=1;

```

```

if(ZY>23) YSay=3;
for(Say=YSay;Say<5;Say++)
    {
        if(ZYPoz==0) eeprom_write(ZY+Say,Zaman[Say]);
        else Zaman[Say]=eeprom_read(ZY+Say);
    }
    lcd_goto(ZYPoz);
}
// Alt menülerin başlangıç değerlerini ekrana yaz - yeni değerleri al hafızaya yaz ---
ZSaatM (char ZSPoz)
{
    unsigned char Hane=1,hane=1,ZPoz=1,Rak_Sec=48;
    static bit Gsaat=0;
    switch(ZSPoz)
        {
case 23:    Zaman[1]=ZamanO(2);
Zaman[2]=ZamanB(2);Zaman[3]=ZamanO(1); Zaman[4]=ZamanB(1);
lcd_goto(ZSPoz=ZSPoz-2); Gsaat=1; break;// Saat anlık değerini yükle
case 86:    Zaman[1]=ZamanO(3); Zaman[2]=ZamanB(3);
Zaman[3]=ZamanO(4); Zaman[4]=ZamanB(4); Zaman[5]=ZamanO(6);
Zaman[6]=ZamanB(6);
lcd_goto(ZSPoz);break;           // Tarih anlık değerini yükle
case 24:ZYukle(24,9); break;      // 1.Yem zamanını Yükle eeprom_read(10)
case 88:ZYukle(88,13); break;// 2.Yem zamanını Yükle eeprom_read(14)
case 21:ZYukle(21,21); break;// Su değiştirme zamanını Yükle eeprom_read(22)
case 26:ZYukle(26,25); Hane=3; break;// Alt Limit değerini Yükle eeprom_read(28)
case 90:ZYukle(90,27); Hane=3;   // Üst Limit değerini Yükle eeprom_read(30)
        }
    for(;;)
        {
            if((Hane==1)&&(hane==1))
                {
                    Rak_Sec=Zaman[1]; hane=0; ZPoz=ZSPoz; lcd_goto(ZPoz);
                }
        }
}

```

```

    }
if((Hane==2)&&(hane==1))
    {
Rak_Sec=Zaman[2]; hane=0; ZPoz=ZSPoz+1; lcd_goto(ZPoz);
    }
if((Hane==3)&&(hane==1))
    {
        Rak_Sec=Zaman[3]; hane=0; ZPoz=ZSPoz+3;
        if((ZSPoz==26)||((ZSPoz==90)) ZPoz=ZSPoz;
        lcd_goto(ZPoz);
    }
if((Hane==4)&&(hane==1))
    {
        Rak_Sec=Zaman[4]; hane=0; ZPoz=ZSPoz+4;
        if((ZSPoz==26)||((ZSPoz==90)) ZPoz=ZSPoz+1;
        lcd_goto(ZPoz);
    }
if((Hane==5)&&(hane==1))
    {
Rak_Sec=Zaman[5]; hane=0; ZPoz=ZSPoz+6; lcd_goto(ZPoz);
    }
if((Hane==6)&&(hane==1))
    {
Rak_Sec=Zaman[6]; hane=0; ZPoz=ZSPoz+7; lcd_goto(ZPoz);
    }
if(Sa_Yem2==0)
    {
        Hane++;
        if((Hane>4)&&(ZSPoz!=86)) goto ZSSon;
        if((Hane>6)&&(ZSPoz==86)) goto ZSSon;
        hane=1; DelayMs(TBekle);
    }
if(So_Yem1==0)

```



```

        {
            if(Hane>1)
                {
                    Hane--;
                    if((Hane<3)&&((ZSPoz==26)||((ZSPoz==90)))) Hane++;
                }
            hane=1;    DelayMs(TBekle);
        }
    if(Y_Aydin==0)
        {
            Rak_Sec++;
            if(Rak_Sec>'9') Rak_Sec=48;
            lcd_goto(ZPoz);    lcd_putch(Rak_Sec); lcd_goto(ZPoz);
            Zaman[Hane]=Rak_Sec; DelayMs(TBekle);
        }
    if(A_Hava==0)
        {
            Rak_Sec--;
            if(Rak_Sec<'0') Rak_Sec='9';
            lcd_goto(ZPoz);    lcd_putch(Rak_Sec);
            lcd_goto(ZPoz);Zaman[Hane]=Rak_Sec; DelayMs(TBekle);
        }
    }
ZSSon:
if (Gsaat==1) ZSPoz=23;
switch(ZSPoz)
    {
        case 23: clkwritebyte(2,(Zaman[1]<<4)+(Zaman[2]&=0x0F));
                clkwritebyte(1,(Zaman[3]<<4)+(Zaman[4]&=0x0F));
                break;                // Saat Ayarı
        case 86:clkwritebyte(3,(Zaman[1]<<4)+(Zaman[2]&=0x0F));
                clkwritebyte(4,(Zaman[3]<<4)+(Zaman[4]&=0x0F));
                clkwritebyte(6,(Zaman[5]<<4)+(Zaman[6]&=0x0F));
    }

```

```

                break;          // Tarih Ayarı
case 24:ZYukle(0,9); break; // 1.Yem zamanı eeprom_write(10,xx)
case 88:ZYukle(0,13); break; // 2.Yem zamanı eeprom_write(13,xx)
case 21:ZYukle(0,21); lcd_goto(27); break; // Su deęiřtirme zamanı
                eeprom_write(21,xx)
case 26:ZYukle(0,25); break;// Alt Limit deęerini Yükle
eeprom_write(30,xx)
                case 90:ZYukle(0,27);// Üst Limit deęerini Yükle eeprom_write(28,xx)
                }
Gsaat=0;
}
// Gün menüsü -----
GunM (unsigned char GPoz)
{
    GMenu_Sec=1; Tkont=1;
    if(GPoz==27)          // Gün deęerini yükle ( su )
        {
            GMenu_Sec = eeprom_read(26); lcd_goto(GPoz);Tkont=1;
        }
    if(GPoz==29)          // Gün deęerini yükle ( zaman )
        {
            GMenu_Sec=clkreadbyte(5); lcd_goto(GPoz=GPoz-2);Tkont=1;
        }
    for(;;)
        {
            if(Y_Aydin==0)    // Yukarı hareket et
                {
                    GMenu_Sec++;
                    if((GMenu_Sec>7)&&(GPoz==27)) GMenu_Sec=1;
                    Tkont=1;
                    DelayMs(TBekle);
                }
            if(A_Hava==0)      // Ařaęı hareket et

```

```

        {
        GMenu_Sec--;
        if((GMenu_Sec<1)&&(GPoz==27)) GMenu_Sec=7;
        Tkont=1;
        DelayMs(TBekle);
        }
    if(Sa_Yem2==0) goto GSon;        // Alt Menüden çık
    if(Tkont>0) GYaz(GPoz,GMenu_Sec);
}

GSon:
if(GPoz==27) eeprom_write(26,GMenu_Sec);// Su değişim Gününü Yaz
if(Hg==1)
    {
        clkwritebyte(5,GMenu_Sec);        // Geçerli Günü Yaz
    }
}

// Alt Menüler -----
ZamanM (void)                // Zaman Menüsü
{
    ZMenu_Sec=1;
    LCD(4,2); LCD(0,21); LCD(1,86);lcd_goto(27);for(Say=0;Say<3;Say++)
    lcd_putchar(gunler[clkreadbyte(5)][Say]);lcd_goto(15);
    for(;;)
        {
            if(T_Menu==0) break;        // Ana ekrana geç
            if(Y_Aydin==0)                // Yukarı hareket et ve
                {                        // Ana Menüye git
                    if(ZMenu_Sec!=0) ZMenu_Sec--;
                    MKay(3,ZMenu_Sec);
                    if(ZMenu_Sec==0) break;
                    DelayMs(TBekle);
                }
            if(A_Hava==0)                // Aşağı hareket et

```

```

        {
            if(ZMenu_Sec<3) ZMenu_Sec++;
            MKay(3,ZMenu_Sec);
            DelayMs(TBekle);
        }
    if(So_Yem1==0) // Alt Menüye git
    {
        switch(ZMenu_Sec)
        {
            case 2: lcd_goto(30); lcd_putch('>'); ZSaatM (23); DelayMs(TBekle); Hg=1;
GunM (29); Hg=0; lcd_goto(30); lcd_putch('<');break; // Saat ayarına git
            case 3: lcd_goto(94); lcd_putch('>'); ZSaatM (86); lcd_goto(94);
lcd_putch('<'); // Tarih ayarına git
        }
        DelayMs(TBekle);
    }
}
    lcd_goto(Menu_Sec);
}
YemM (void) // Yem Menüsü
{
    YMenu_Sec=1; LCD(5,3);
    lcd_goto(24); lcd_putch(eeprom_read(10)); lcd_putch(eeprom_read(11));
lcd_putch(':'); // 1.Yem atma zamanını yazdır
    lcd_putch(eeprom_read(12));lcd_putch(eeprom_read(13));
    lcd_goto(88); lcd_putch(eeprom_read(14)); lcd_putch(eeprom_read(15));
    lcd_putch(':'); // 2.Yem atma zamanını yazdır
    lcd_putch(eeprom_read(16));lcd_putch(eeprom_read(17));lcd_goto(15);
    for(;;)
    {
        if(T_Menu==0) break; // Ana ekrana geç
        if(Y_Aydin==0) // Yukarı hareket et ve
        { // Ana Menüye git

```

```

        if(YMenu_Sec!=0) YMenu_Sec--;
        MKay(3,YMenu_Sec);
        if(YMenu_Sec==0) break;
        DelayMs(TBekle);
    }
    if(A_Hava==0) // Aşağı hareket et
    {
        if(YMenu_Sec<3) YMenu_Sec++;
        MKay(3,YMenu_Sec);
        DelayMs(TBekle);
    }
    if(So_Yem1==0) // Alt Menüye git
    {
        switch(YMenu_Sec)
        {
            case 2: lcd_goto(30); lcd_putch('>'); ZSaatM (24);
lcd_goto(30); lcd_putch('<');break; // 1.Yem ayarına git
            case 3: lcd_goto(94); lcd_putch('>'); ZSaatM (88);
lcd_goto(94); lcd_putch('<');break; // 2.Yem ayarına git
        }
        DelayMs(TBekle);
    }
}

IsiM (void) // Isı Menüsü
{
    IMenu_Sec=1; LCD(6,1);
    lcd_goto(26); lcd_putch(eeprom_read(28)); lcd_putch(eeprom_read(29));
        // Üst ısı limitini yazdır
    lcd_goto(90); lcd_putch(eeprom_read(30)); lcd_putch(eeprom_read(31));
    lcd_goto(15); // Alt ısı limitini yazdır
    for(;;)
        {

```

```

if(T_Menu==0) break;           // Ana ekrana geç
if(Y_Aydin==0)                 // Yukarı hareket et ve
{
    // Ana Menüye git
    if(IMenu_Sec!=0) IMenu_Sec--;
    MKay(4,IMenu_Sec);
    if(IMenu_Sec==0) break;
    DelayMs(TBekle);
}
if(A_Hava==0)                  // Aşağı hareket et
{
    if(IMenu_Sec<3) IMenu_Sec++;
    MKay(4,IMenu_Sec);
    DelayMs(TBekle);
}
if(So_Yem1==0)                // Alt Menüye git
{
    switch(IMenu_Sec)
    {
        case 2: lcd_goto(30); lcd_putch('>'); ZSaatM (26);
lcd_goto(30); lcd_putch('<');break;           // Üst Limit ısı ayarına git
        case 3: lcd_goto(94); lcd_putch('>'); ZSaatM (90);
lcd_goto(94); lcd_putch('<');           // Alt Limit ısı ayarına git
    }
    DelayMs(TBekle);
}
}

SuM (void)                    // Su Menüsü
{
    SMenu_Sec=1;
    LCD(7,2);
    lcd_goto(21); lcd_putch(eeprom_read(22)); lcd_putch(eeprom_read(23));
    lcd_putch(':');           // Su değiştirme zamanını yazdır
}

```

```

lcd_putchar(eeprom_read(24));lcd_putchar(eeprom_read(25));
lcd_goto(27);for(Say=0;Say<3;Say++)
lcd_putchar(gunler[eeprom_read(26)][Say]);lcd_goto(15);
for(;;)
    {
        if(T_Menu==0) break;           // Ana ekrana geç
        if(Y_Aydin==0)                 // Yukarı hareket et ve
            {                           // Ana Menüye git
                if(SMenu_Sec!=0) SMenu_Sec--;
                MKay(5,SMenu_Sec);
                if(SMenu_Sec==0) break;
                DelayMs(TBekle);
            }
        if(A_Hava==0)                 // Aşağı hareket et
            {
                if(SMenu_Sec<2) SMenu_Sec++;
                MKay(5,SMenu_Sec);
                DelayMs(TBekle);
            }
        if((So_Yem1==0)&&(SMenu_Sec==2)) // Su Değişirme saatine git
            {
                lcd_goto(30); lcd_putchar('>');
                ZSaatM (21); DelayMs(TBekle);
                GunM (27);
                lcd_goto(30); lcd_putchar('<');
                DelayMs(TBekle);
            }
    }
}
// Ana Menü -----
AMenu (void)
{
    LCD(3,1); lcd_goto(15); Menu_Sec=1;

```

```

for(;;)
{
    if(Y_Aydin==0)           // Yukarı hareket et ve
    {
        // Ana Ekranı git
        if(Menu_Sec!=0) Menu_Sec--;
        MKay(1,Menu_Sec);
        if(Menu_Sec==0) break;
        DelayMs(TBekle);
    }
    if(A_Hava==0)           // Aşağı hareket et
    {
        if(Menu_Sec<4) Menu_Sec++;
        MKay(1,Menu_Sec);
        DelayMs(TBekle);
    }
    if(So_Yem1==0)         // Alt Menüye git
    {
        switch(Menu_Sec)
        {
            case 1: ZamanM (); LCD(3,1); lcd_goto(15); break; // Zaman ayarına git
            case 2: YemM (); LCD(3,1); lcd_goto(79); break; // Yem ayarına git
            case 3: IsiM (); LCD(3,1); lcd_goto(31); break; // Isı Limitlerine git
            case 4: SuM (); LCD(3,1); lcd_goto(95); // Su Değişimine git
        }
        DelayMs(TBekle);
    }
    if(T_Menu==0)         // Ana ekrana geç
    {
        DelayMs(TBekle);
        break;
    }
}
}

```



```

// Tuş Kontrolleri -----
TusK (void)
{
    if(T_Menu==0)                // Menüye Geç
    {
        DelayMs(TBekle); AMenu(); ETazele();
    }

    if(eeprom_read(0)==0)
    {
        if(So_Yem1==0)           // 1. Yemlikten yem at
        {
            DKon(1);goto tt;
        }
        if(Sa_Yem2==0)           // 2. Yemlikten yem at
        {
            DKon(2);goto tt;
        }
        if(A_Hava==0)            // Hava motorunu Aç/Kapat
        {
            if(eeprom_read(2)==0) // Hava Açıkysa, kapat
            {
                DKon(4);goto tt;
            }
            else                  // Hava Kapalıysa, Aç
            {
                DKon(4);goto tt;
            }
        }
        if(Y_Aydin==0)           // Aydınlatmayı Aç/Kapat
        {
            DKon(3);
        }
    }
}

```

```

    }
tt:asm("NOP");
}
// Ana program *****
void main(void)
{
INTCON =PIR1 =PIE1 = 0;//perpharial interrupts are disable
//--- Analog Girişleri düzenle
    ADCON1=0b10001110;    //RA0 analog
//--- Dijital Giriş - Çıkışları düzenle
//  Dijital portlar için 0 = Çıkış    1 = Giriş
//    76543210
    TRISA=0b00100101;
    TRISB=0b10000001;
    TRISC=0b11110000;
    TRISD=0b11111100;
    TRISE=0b00000010;
    PORTA=0b00000000;
    PORTB=0b00001000;
    PORTC=0b00000000;
    PORTD=0b00000000;
    PORTE=0b00000000;
    initialize_DS1302();
    lcd_init();    lcd_clear();
    lcd_goto(4);lcd_puts("Baslangic");
    lcd_goto(66); lcd_puts("Ayarlamalari");
    lcd_goto(19); lcd_puts("Yapiliyor");
    lcd_goto(84); lcd_puts("Bekleyin.");
// ilk çalıştırmada eeprom FF oluyorsa kullanılan yerleri sıfırla
    for(Say=0;Say<=9;Say++)
        {
            Zkont=eeprom_read(Say);
            if(Zkont>1)

```

```
        {
            DelayMs(10);eeprom_write(Say,0);DelayMs(10);
        }
    }
for(Say=10;Say<=25;Say++)
    {
        Zkont=eeprom_read(Say); DelayMs(5);
        if((Zkont<48)||(Zkont>57))
            {
                DelayMs(10); eeprom_write(Say,'0'); DelayMs(10);
            }
    }
if(eeprom_read(26)>7)
    {
        DelayMs(10); eeprom_write(26,1); DelayMs(10);
    }
if(eeprom_read(27)>2)
    {
        DelayMs(10); eeprom_write(27,1); DelayMs(10);
    }
for(Say=28;Say<35;Say++)
    {
        Zkont=eeprom_read(Say);
        if((Zkont<48)||(Zkont>57))
            {
                DelayMs(10); eeprom_write(Say,'0'); DelayMs(10);
            }
    }
BSes(3,10,5);
ETazele();
for(;;)
    {
        Yenile();
    }
```

```

        TusK ();
        if(IR==0) U_Kum();
//    U_Kum();
        }
}

```

## A.2. Basit Ses Kontrolleri

```

/*****
*          Basit Ses kontrolleri          *
*****/
#include    <pic.h>
#include    "BSes.h"
#include    "delay.h"
void BSes(unsigned char Tekrar,unsigned char Ses,unsigned char Dur)
{
    unsigned char tek=0;
    for(tek=1;tek<=Tekrar;tek++)
        {
            Buzz=1;
            DelayMs(Ses);
            DelayMs(Ses);
            DelayMs(Ses);
            DelayMs(Ses);
            Buzz=0;
            DelayMs(Dur);
            DelayMs(Dur);
            DelayMs(Dur);
            DelayMs(Dur);
        }
}

```

### A.3. Delay

```

/*    Delay functions
 *    See delay.h for details
 *    Make sure this code is compiled with full optimization!!! */
#include    "delay.h"
void
DelayMs(unsigned char cnt)
{
#if    XTAL_FREQ <= 2MHZ
    do {
        DelayUs(996);
    } while(--cnt);
#endif
#if    XTAL_FREQ > 2MHZ
    unsigned char i;
    do {
        i = 4;
        do {
            DelayUs(250);
        } while(--i);
    } while(--cnt);
#endif
}

```

### A.4. Dış Kontroller

```

/*****
 *
 *                Dış Kontroller                *
 *****/
#include <pic.h>
#include "DKontrol.h"
#include "LCDGosterge.h"

```

```

#include "delay.h"
#include "Ses.h"
#include "lcd.h"
#include "ETazele.h"
#include <stdlib.h>
unsigned char
EKZ[3],EKont,EKont1,EKont2,Saat,Dakika,EKsaat,EKsaat1,EKdakika,Hata;
UzunBekle(void)
{
    static bank2 unsigned char Bekle=0;
    for(Bekle=0;Bekle<8;Bekle++) DelayMs(250);
}
void DKon(unsigned char DKontrol)
{
    if(DKontrol==0)                // Enerji Kontrolü
    {
        if(EKes==1)
        {
            lcd_goto(80);
            lcd_puts(" Enerji Kesildi.");
            EKZ[0]=ZamanO(2);EKZ[1]=ZamanB(2);
            Saat=atoi(EKZ);    // Geçerli Saat
            EKZ[0]=ZamanO(1);EKZ[1]=ZamanB(1);
            Dakika=atoi(EKZ); // Geçerli Dakika
            EKont2=0;
            if(EKont==0)// Enerji kesildi bekleme zamanını hesapla
            {
                BSes(5,15,5);
                HAVA=0;    // Havalandırmayı kapat
            }
            EKont=1;
            EKdakika=Dakika;
            EKsaat=Saat+5;
            if (Saat>=19) EKsaat=abs((24-Saat)-5);
        }
    }
}

```

```

eeprom_write(0,1);DelayMs(5);// Elektrik kesildi. eeproma yaz
        UzunBekle();
        ETazele();
    }
    if((EKsaat==Saat)&&(EKdakika==Dakika))
    {
        if(EKont1==0) // Enerji kesildi bekledikten sonra
        {
            // 5 dakika çalışma zamanını hesapla
            HAVA=1;// Havalandırma Çalışıyor.
            EKont1=1;
            // EKsaat1=Saat+1;
            EKsaat1=Saat;
            EKdakika=Dakika+5;
            // if (Saat==23) EKsaat1=0;
            if (Dakika>=55) EKdakika=60-Dakika;
        }
    }
    if((EKsaat1==Saat)&&(EKdakika==Dakika)) // 5 dakika doldu
    {
        EKont=0; // herşeyi sıfırla
        EKont1=0;
        EKsaat=0;
        EKsaat1=0;
    }
}
if(EKes==0)
{
    if(EKont2==0)
    {
        eeprom_write(0,0); // Elektrik geldi. eeproma yaz
        BSes(5,3,5);
        if(EEPROM_READ(2)==1) HAVA=1;// Havalandırmayı Çalıştır.
    }
}

```

```

else HAVA=0;      // Havalandırmayı Kapat.
    EKont=0;
    EKont1=0;
    EKont2=1;
    UzunBekle();
    ETazele();
}
}
DelayMs(5);
goto Son;
}
if(DKontrol==1)      // 1.Yemlik
{
    LCD(11,80);
    BSes(1,5,0);
    YM1=1;
    DelayMs(250);
    YM1=0;
}
if(DKontrol==2)      // 2. Yemlik
{
    LCD(11,80);
    BSes(1,5,0);
    YM2=1;
    DelayMs(250);
    YM2=0;
}
if(DKontrol==3)      // Aydınlatma
{
    if(EEPROM_READ(3)==0)
    {
        BSes(1,5,5);
        eeprom_write(3,1);
    }
}

```



```

        AYDIN=1;           // Aç
        goto Son;
    }

    BSes(2,5,5);
    eeprom_write(3,0);
    AYDIN=0;           // Kapat
}

if(DKontrol==4)           // Havalandırma Motoru
{
    if(EEPROM_READ(2)==0)
    {
        BSes(1,5,5);
        eeprom_write(2,1);
        HAVA=1;           // Aç
        goto Son;
    }

    BSes(2,5,5);
    eeprom_write(2,0);
    HAVA=0;           // Kapat
    goto Son;
}

if(DKontrol==5)           // Su Değiştir
{
    Bosalt:

        LCD(12,80);
        BSes(1,5,0);
        if(ASev==0)
        {
            Hata=1;
            goto Doldur;
        }

        ADSP=1;           // Dip Süpürülüyor.
        DelayMs(100);     // Süpürme için bekle

```

```

while(ASev==1) AKSV=1;      // Kirli Su Valfini Aç
ADSP=0;                    // Dip Süpürmeyi bitir.
AKSV=0;                    // Kirli Su Valfini Kapat

```

Doldur:

```

LCD(13,80);
BSes(2,5,5);
while(ASev==0) ATSV=1;    // Temiz Su Valfini Aç
ATSV=0;                   // Temiz Su Valfini Kapat
if(Hata==1)
{
    Hata=0;
    goto Bosalt;
}
}

```

Son:

```

DelayMs(5);
if(DKontrol!=0) ETazele();
DelayMs(5);
}

```

## A.5. DS1302

```

/*****
/*          DS1302.C          */
*****/

#include "stdio.h"
#include "pic168xa.h"      /* Register declarations for pic16f877 */
#include "DS1302.h"
/* ----- */

void reset_3w()
{
    SCLK = 0;

```

```

RSTB = 0;
RSTB = 1;
}
/* ----- */
void wbyte_3w(char W_Byte)
{
    char i;
    IO_TRIS = 0;
    for(i = 0; i < 8; ++i)
    {
        if(W_Byte & 0x01) // eğer datanın ilgili '1' ise
        {
            IO = 1;      //çıkışı '1' yap
        }
        else IO = 0;      //yoksa 0 yap
        SCLK = 0;
        SCLK = 1;
        W_Byte >>= 1;
    }
}
/* ----- */
char rbyte_3w()
{
    char i;
    char R_Byte;
    char TmpByte;
    R_Byte = 0x00;
    IO_TRIS = 1;
    for(i=0; i<8; ++i)
    {
        SCLK = 1;
        SCLK = 0;
        TmpByte = (char)IO;
    }
}

```

```

        TmpByte <<= 7;
        R_Byte >>= 1;
        R_Byte |= TmpByte;
    }
    return R_Byte;
}
/* ----- */
void clkwritebyte(char ClkAdd,char ClkData)
{
    /* Prepare Command Byte for Clock Write */
    ClkAdd = ((ClkAdd * 2) | 0x80);
    reset_3w();
    wbyte_3w(0x8e);    /* control register */
    wbyte_3w(0);      /* disable write protect */
    reset_3w();
    wbyte_3w(ClkAdd);
    wbyte_3w(ClkData);
    reset_3w();
    wbyte_3w(0x8e);    /* control register */
    wbyte_3w(0x80);    /* disable write protect */
}
// -----
void ramwritebyte(char ramadd,char ramdata)
{
    // Prepare Command Byte for Ram Write
    ramadd = ((ramadd * 2) | 0xC0);
    reset_3w();
    wbyte_3w(ramadd);
    wbyte_3w(ramdata);
    reset_3w();
}
void burstramrd()    // ----- read RAM in burst mode -----
{

```

```

char rmary[31];
char i;
    reset_3w();
    wbyte_3w(0xFF);    // RAM burst read
    for (i=0; i<31; ++i)
    {
        rmary[i] = rbyte_3w();
    }
    reset_3w();
}
void burstramwr(char ramdata)    // ----- write one value to entire array -----
{
char i;
    reset_3w();
    wbyte_3w(0xfe);    // RAM burst write
    for (i=0; i<31; ++i)
    {
        wbyte_3w(ramdata);
    }
    reset_3w();
}
void initialize_DS1302()    /* ----- */
/* Note: NO error checking is done on the user entries! */
{
//char yr, mn, date, dy, hr, min, sec;//, day;

    reset_3w();
    wbyte_3w(0x8e);    /* control register */
    wbyte_3w(0);    /* disable write protect */
    reset_3w();
    wbyte_3w(0x90);    /* trickle charger register */
    wbyte_3w(0xab);    /* enable, 2 diodes, 8K resistor */
    reset_3w();
}

```

```

        wbyte_3w(0xbe);          /* clock burst write (eight registers) */
//    wbyte_3w(sec);
//    wbyte_3w(min);
//    wbyte_3w(hr);
//    wbyte_3w(date);
//    wbyte_3w(mn);
//    wbyte_3w(dy);
//    wbyte_3w(yr);
        wbyte_3w(0);          /* must write control register in burst mode */
        reset_3w();
    }
char read_clk_burst() // ---- read and display clock registers ----
{
//uchar lsec = 99, mon, day, yr, sec, hrs, dte, min;
char lsec = 99, mon, day, yr, sec, hrs, dte, min;
        reset_3w();
        wbyte_3w(0xBF);    // clock burst
        sec = rbyte_3w();
        min = rbyte_3w();
        hrs = rbyte_3w();
        dte = rbyte_3w();
        mon = rbyte_3w();
        day = rbyte_3w();
        yr = rbyte_3w();
        reset_3w();
}
char clkreadbyte(char ClkAdd)
{
    /* Prepare Command Byte for Clock Write */
    char Clkdata;
    ClkAdd = (ClkAdd * 2) | 0x81;
    reset_3w();
    wbyte_3w(ClkAdd);
}

```

```

    Clkdata = rbyte_3w();
    reset_3w();
    return (Clkdata);
}

```

## A.6. Ekran Tazeleme

```

/*****
*           Ekran Tazeleme           *
*****/

#include    <pic.h>
#include    "LCDGosterge.h"
#include    "lcd.h"
ETazele(void)                                // Ana Ekranı Yazdır
{
    lcd_clear();
    LCD(0,0);                                // Saat Yazdır
    lcd_goto(5);
    lcd_puts(" | ");
    LCD(1,8);                                // Tarih Yazdır
    LCD(2,64);                                // Isı Yazdır
    lcd_goto(69);
    lcd_puts(" | ");
    LCD(17,71);                               // Haftanın Gününü Yazdır
    LCD(10,16);                               // Kontroller yazdır
    if(eeprom_read(3)==1)                    // Aydınlatma Açık
        {
            lcd_goto(25);
            lcd_puts("+");
        }
    else                                      // Aydınlatma Kapalı
        {
            lcd_goto(25);
            lcd_puts("-");
        }
}

```

```

    }
    if(eeprom_read(2)==1)                // Havalandırma Açık
    {
        lcd_goto(28);
        lcd_puts("+");
    }
    else                                  // Havalandırma Kapalı
    {
        lcd_goto(28);
        lcd_puts("-");
    }
    if(eeprom_read(0)==0)                // Enerji kesildi
    {
        lcd_goto(80);
        lcd_puts(" Enerji Kesildi.");
    }
    LCD(8,80);                            // Sistem Normal
}

```

## A.7. Lcd

```

/*
 * LCD interface example
 * Uses routines from delay.c
 * This code will interface to a standard LCD controller
 * like the Hitachi HD44780. It uses it in 4 bit mode, with
 * the hardware connected as follows (the standard 14 pin
 * LCD connector is used):
 *
 * PORTC bits 0-3 are connected to the LCD data bits 4-7 (high nibble)
 * PORTD bit 0 is connected to the LCD RS input (register select)
 * PORTD bit 1 is connected to the LCD EN bit (enable)
 *

```



```

*      To use these routines, set up the port I/O (TRISA, TRISB) then
*      call lcd_init(), then other routines as required.
*/

#include    <pic.h>
#include    "lcd.h"
#include    "delay.h"

static bit LCD_RS    @ ((unsigned)&PORTD*8+1);    // Register select
static bit LCD_EN    @ ((unsigned)&PORTD*8+0);    // Enable
#define     LCD_STROBE    ((LCD_EN = 1),(LCD_EN=0))
/* write a byte to the LCD in 4 bit mode */
void
lcd_write(unsigned char c)
{
    PORTC = (PORTC & 0xF0) | (c >> 4);
    LCD_STROBE;
    PORTC = (PORTC & 0xF0) | (c & 0x0F);
    LCD_STROBE;
    DelayUs(40);
}
/*
*      Clear and home the LCD
*/
void
lcd_clear(void)
{
    LCD_RS = 0;
    lcd_write(0x1);
    DelayMs(2);
}
/* write a string of chars to the LCD */
void
lcd_puts(const char * s)
{

```

```

        LCD_RS = 1; // write characters
        while(*s)
            lcd_write(*s++);
    }
    /* write one character to the LCD */
    void
    lcd_putchar(char c)
    {
        LCD_RS = 1; // write characters
        PORTC = (PORTC & 0xF0) | (c >> 4);
        LCD_STROBE;
        PORTC = (PORTC & 0xF0) | (c & 0x0F);
        LCD_STROBE;
        DelayUs(40);
    }
    /*
    * Go to the specified position
    */
    void
    lcd_goto(unsigned char pos)
    {
        LCD_RS = 0;
        lcd_write(0x80+pos);
    }
    /* initialise the LCD - put into 4 bit mode */
    void
    lcd_init(void)
    {
        LCD_RS = 0; // write control bytes
        DelayMs(15); // power on delay
        PORTC = 0x3; // attention!
        LCD_STROBE;
        DelayMs(5);
    }

```

```

LCD_STROBE;
DelayUs(100);
LCD_STROBE;
DelayMs(5);
PORTC = 0x2;      // set 4 bit mode
LCD_STROBE;
DelayUs(40);
lcd_write(0x28);  // 4 bit mode, 1/16 duty, 5x8 font
lcd_write(0x08);  // display off
lcd_write(0x0F);  // display on, blink cursor on
lcd_write(0x06);  // entry mode
}

```

## A.8. LCD Mesajları

```

/*****
*          LCD Mesajları          *
*****/

#include <pic.h>
#include <stdlib.h>
#include <stdio.h>
#include "lcd.h"
#include "ds1302.h"
#include "LCDGosterge.h"
#include "BSes.h"
#include "delay.h"
//#include "ETazele.h"

unsigned char onlar,birler,zaman,Kontrol,hg,Eg,x,T0,T1,T2,Tmp[3],isi[4]="000";
static bank2 unsigned char Gunler[8][11] = {" ", " Pazar ", "Pazartesi", " Sali ", "
Carsamba", " Persembe", " Cuma ", "Cumartesi"};

// DS1302 -----
int ZamanO(unsigned char ZOnlar)          // Zaman Onlar basamağı
{

```

```

        zaman=clkreadbyte(ZOnlar); onlar=zaman>>4; onlar|=0x30; return(onlar);
    }
int ZamanB(unsigned char ZBirler)          // Zaman Birler basamağı
{
    birler=clkreadbyte(ZBirler); birler&=0x0F; birler|=0x30; return(birler);
}
// LM35 -----
LM35(void)                                // Isı değeri
{
    ADCON0=0x41;
    ADCON0=0x45;
    while((ADCON0&4)!=0);
    sprintf(isi,"%d",(int)((ADRESH+ADRESL)*49)/10);
//    sprintf(isi,"%d",(int)(((ADRESH+ADRESL)*256.0)*4.8828125)/100);
}
LCD(unsigned char LCDms,unsigned char LCDPos)
{
    switch (LCDms)                          // LCD Mesajlarını Seç
    {
        case 0: lcd_goto(LCDPos);
            lcd_putchar(ZamanO(2)); // Saat Onlar
            lcd_putchar(ZamanB(2)); // Saat Birler
            lcd_putchar(':');
            lcd_putchar(ZamanO(1)); // Dakika Onlar
            lcd_putchar(ZamanB(1)); // Dakika Birler
            break;
        case 1: lcd_goto(LCDPos);
            lcd_putchar(ZamanO(3)); // Gün Onlar
            lcd_putchar(ZamanB(3)); // Gün Birler
            lcd_putchar(':');
            lcd_putchar(ZamanO(4)); // Ay Onlar
            lcd_putchar(ZamanB(4)); // Ay Birler
            lcd_putchar(':');
    }
}

```

```

        lcd_putchar(ZamanO(6)); // Yil Onlar
        lcd_putchar(ZamanB(6)); // Yil Birler
        break;
case 2: LM35(); // Isı
        lcd_goto(LCDPos);
        lcd_putchar('+');
        lcd_goto(LCDPos+1);
        lcd_putchar(isi[0]);
        lcd_putchar(isi[1]);
        lcd_putchar(223);lcd_putchar('C');
        lcd_goto(96);
        eeprom_write(32,isi[1]);
        eeprom_write(33,isi[0]);
        DelayMs(5);
        break;
case 3: lcd_clear(); // Menü
        lcd_goto(LCDPos);
        lcd_puts("Saat-Tarih ..<^");
        lcd_goto(65);
        lcd_puts("Yemleme .....<|");
        lcd_goto(17);
        lcd_puts("Isi Limitleri<|");
        lcd_goto(81);
        lcd_puts("Su Degisimi..<|");
        break;
case 4: lcd_clear(); // Zaman Ayarı
        lcd_goto(LCDPos);
        lcd_puts("Zaman Ayari ^");
        lcd_goto(64);
        lcd_puts("-----");
        lcd_goto(16);
        lcd_puts("Saat= <|");
        lcd_goto(80);

```

```

        lcd_puts("Tarih= <|");
        lcd_goto(15);
        break;
case 5: lcd_clear();                // Yem Ayarı
        lcd_goto(LCDPos);
        lcd_puts("Yem Ayari ^");
        lcd_goto(64);
        lcd_puts("-----");
        lcd_goto(16);
        lcd_puts("1.Yem = <|");
        lcd_goto(80);
        lcd_puts("2.Yem = <|");
        lcd_goto(15);
        break;
case 6: lcd_clear();                // Isı Limitleri
        lcd_goto(LCDPos);
        lcd_puts("Isi Limitleri ^");
        lcd_goto(64);
        lcd_puts("-----");
        lcd_goto(16);
        lcd_puts("Ust Lim.=+
");lcd_putchar(223);lcd_puts("C<|");
        lcd_goto(80);
        lcd_puts("Alt Lim.=+
");lcd_putchar(223);lcd_puts("C<|");
        break;
case 7: lcd_clear();                // Su Değişim Ayarları
        lcd_goto(LCDPos);
        lcd_puts("Su Degisimi ^");
        lcd_goto(64);
        lcd_puts("-----");
        lcd_goto(16);
        lcd_puts("Su = <<");

```

```

        lcd_goto(15);
        break;
case 8: if((hg==0)&&(eeprom_read(0)==0)) // Sistem Normal LCD(8,80);
        {
            lcd_goto(LCDPos);
            lcd_puts("-Sistem Normal-");
        }
        break;
case 10:lcd_goto(LCDPos); // Kontoller LCD(10,16);
        lcd_puts("---ph | A H I+");
        break;
case 11:lcd_goto(LCDPos); // Yem Attıyorum LCD(11,80);
        lcd_puts(" Yem Atıyorum. ");
        break;
case 12:lcd_goto(LCDPos); // Su Boşaltıyorum LCD(12,80);
        lcd_puts("Su Bosaltıyorum.");
        break;
case 13:lcd_goto(LCDPos); // Su Dolduruyorum LCD(13,80);
        lcd_puts("Su Dolduruyorum.");
        break;
case 17:lcd_goto(LCDPos); // Haftanın Gününü Yazdır LCD(17,xx);
        for(x=0;x<9;x++)
            {
                lcd_putchar(Gunler[clkreadbyte(5)][x]);
            }
        lcd_goto(96);
        break;
case 18: // LCD(18,80);
        T0=atoi(isi); // T0 'a <-- Okunan sıcaklığı yükle
        Tmp[0]=eeprom_read(30); // Isı Alt Limit Onlar
        Tmp[1]=eeprom_read(31); // Isı Alt Limit Birler
        T1=atoi(Tmp); // T1 'e <-- alt sıcaklık limitini yükle
        Tmp[0]=eeprom_read(28); // Isı Üst Limit Onlar

```

```

    Tmp[1]=eeprom_read(29);          // Isı Üst Limit Birler
    T2=atoi(Tmp);                 // T2 'ye <-- Üst sıcaklık limitini yükle
    if(T0>T2)
        {
        lcd_goto(LCDPos); // Isı Çok Yüksek
        lcd_puts("Isi Cok Yuksek !");
        if(hg!=2)
            {
                BSes(5,10,5);

                hg=2;
            }
        ISI=0;
        lcd_goto(31);
        lcd_puts("-");
        eeprom_write(4,0);

        }
    if(T0<T1)
        {
        lcd_goto(LCDPos); // Isı Çok Düşük
        lcd_puts(" Isi Cok Dusuk !");
        if(hg!=1)
            {
                BSes(20,5,5);
                hg=1;
            }
        ISI=1;
        eeprom_write(4,1);
        lcd_goto(31);
        lcd_puts("+");

        }
    if((T1<T0)&&(T0<T2)) // Isı normal
        {

```



```

//
ISI=1;
eeprom_write(4,1);
hg=0;
lcd_goto(31);
lcd_puts("+");
}

break;
}
}

```

## A.9. Uzaktan Kumanda kontrolleri

```

/*****
*      Uzaktan Kumanda kontrolleri      *
*****/

#include <pic.h>
#include "UKumanda.h"
#include "delay.h"
#include "DKontrol.h"
void U_Kum(void)
{
static bank1 unsigned char Pals_Say=0,Kumanda[30];
unsigned int Boy=0;
/-- Başlık kısmı tespit et ve alta, Aradan girişi atla.
A1:      Boy++;
          if (Boy>2000) goto A6;
          if (IR==1) goto A1;      // Birleri atla
          if (Boy>500)             // Kodun başını buldun
              {
                  Boy=0;
A0:      Boy++;
          if (Boy>500) goto A6;

```

```

if (IR==0) goto A0;           // Sıfırları atla
Boy=0;
goto A2;
    }
Boy=0;
AA:   Boy++;
      if (Boy>1000) goto A6;
      if (IR==0) goto AA;
      Boy=0;
      goto A1;

//--- Kodu almaya başla
A2:   if (IR==1)
      {
          Boy=0;
          Kumanda[Pals_Say++]='1';
A3:   Boy++;
          if (Boy>100) goto A6;
          if (IR==1) goto A3;
          Boy=0;
      }
      if (IR==0)
      {
A4:   Kumanda[Pals_Say++]='0';
          if (IR==0)
          {
              Boy++;
          }
          if (Boy>50)
          {
              Boy=0;
              Kumanda[Pals_Say++]='0';
              goto A5;
          }
      }

```

```

        goto A4;
    }
A5:        Boy++;
        if ((IR==0)&&(Boy<100)) goto A5;
        Boy=0;
    }
if (Pals_Say>21)
{
    //--- P + ( Aydınlatma )
    if ((Kumanda[9]=='0')&&(Kumanda[10]=='0')
    &&(Kumanda[16]=='0')&&(Kumanda[17]=='0'))
        {
            DKon(3);goto A6;// Aydınlatmayı Aç/Kapat
        }
    //--- P - ( Hava )
    if((Kumanda[1]=='0')&&(Kumanda[2]=='0')&&
    (Kumanda[10]=='0')&&(Kumanda[11]=='0')&&
    (Kumanda[17]=='0')&&(Kumanda[18]=='0'))
        {
            if(eeprom_read(2)==0)// Hava motorunu Kapat
                {
                    DKon(4);goto A6;
                }
            else // Hava motorunu Aç
                {
                    DKon(4);goto A6;
                }
        }
    //--- + ( Yemleme2 )
    if((Kumanda[3]=='0')&&(Kumanda[4]=='0')&&(Kumanda[10]=='0')
    &&(Kumanda[11]=='0')&&(Kumanda[17]=='0')&&(Kumanda[18]=='0'))
        {

```

```

        DKon(2);goto A6; // 1. Yemlikten yem at
        }
        //--- - ( Yemleme1 )
        if((Kumanda[1]=='0')&&(Kumanda[2]=='0')&&(Kumanda[4]=='0')&&
        (Kumanda[5]=='0')&&(Kumanda[11]=='0')&&(Kumanda[12]=='0')&&
        (Kumanda[18]=='0')&&(Kumanda[19]=='0'))
        {
            DKon(1);goto A6; // 2. Yemlikten yem at
            }
A6:      for(Pals_Say=0;Pals_Say<29;Pals_Say++)
        {
            Kumanda[Pals_Say]=' ';
            }
        Pals_Say=0;
        Boy=0;
        DelayMs(25);
        }
        else goto A2;
    }

```

## EK B. HEADER (H) DOSYALARI

### B.1. Ana Program

```

/*****
*      Değişken Tanımlamaları      *
*****/

__CONFIG(WDTDIS&XT&PWRTEN&BORDIS&LVPDIS&DEBUGDIS);
#undef XTAL_FREQ
#define XTAL_FREQ 4MHZ
//#define IR      RD7          // IR girişi
unsigned char Say=0;
unsigned char TBekle=80;
unsigned char Menu_Sec=0;
unsigned char ZMenu_Sec=0;
unsigned char YMenu_Sec=0;
unsigned char IMenu_Sec=0;
unsigned char SMenu_Sec=0;
unsigned char GMenu_Sec=0;
unsigned char Tkont=1,Zkont=0,Skont=0,Ekont=0,Hg=0;
static bank1 unsigned char gunler[8][4] = {
", "Paz", "Pts", "Sal", "Car", "Per", "Cum", "Cts"};
static bank1 unsigned char Zaman[7]={48,48,48,48,48,48,48};
#define Y_Aydin      RC5
#define      A_Hava      RD5
#define      So_Yem1      RD3
#define      Sa_Yem2      RD2
#define      T_Menu      RD4
#define EKes RA5          // Şebeke Kontrol
#define ISI      RB3
/*
RA0 - LM35 (1s1)
RA1 - Yem1M

```

RA2 - IR  
RA3 - Yem2M  
RA4 - Ses  
RA5 - Enerji Kesildi  
RB0 -  
RB1 - Hava Motoru  
RB2 - Aydınlatma  
RB3 - Isıtıcı  
RB4 - ( Akvaryum Dip Süpürme ) ADSPompası  
RB5 - ( Akvaryum Temiz Su ) ATSValfi  
RB6 - ( Akvaryum Kirli Su ) AKSValfi  
RB7 -  
RC0 - LCD-S1  
RC1 - LCD-S2  
RC2 - LCD-S3  
RC3 - LCD-S4  
RC4 - Akvaryum Su Seviyesi  
RC5 - Yukarı/Aydınlatma  
RC6 - TX  
RC7 - RX  
RD0 - LCD-E  
RD1 - LCD-RS  
RD2 - Sol/Yem1  
RD3 - Sağ/Yem2  
RD4 - Tamam  
RD5 - Aşağı/Hava  
RD6 -  
RD7 -  
RE0 - SCLK  
RE1 - I/O  
RE2 - RST  
LCD satır adresleri  
-----

1.Satır - 0

2.Satır - 64

3.Satır - 16

4.Satır - 80

Saat Yerleşimi

-----

0- Saniye

1- Dakika

2- Saat

3- Gün

4- Ay

5- Haftanın Günü

6- Yıl

7-

8- Şarj

9-

Eeprom yerleşimi

-----

eeprom 0 // Elektrik Kesildi 0 = Kesildi 1 = Var

eeprom 1 // Su devir daim .. 0 = Duruyor 1 = Çalışıyor

eeprom 2 // Havalandırma ... 0 = Kapalı 1 = Açık

eeprom 3 // Aydınlatma ..... 0 = Kapalı 1 = Açık

eeprom 4 // Isıtıcı ..... 0 = Kapalı 1 = Açık

eeprom 5 //

eeprom 6 //

eeprom 7 //

eeprom 8 //

eeprom 9 //

eeprom 10 // Yem1 Saat Onlar

eeprom 11 // Yem1 Saat Birler

eeprom 12 // Yem1 Dakika Onlar

eeprom 13 // Yem1 Dakika Birler

eeprom 14 // Yem2 Saat Onlar

```

eeprom 15 // Yem2 Saat Birler
eeprom 16 // Yem2 Dakika Onlar
eeprom 17 // Yem2 Dakika Birler
eeprom 18
eeprom 19
eeprom 20
eeprom 21
eeprom 22 // Su Saat Onlar
eeprom 23 // Su Saat Birler
eeprom 24 // Su Dakika Onlar
eeprom 25 // Su Dakika Birler
eeprom 26 // Su Değişme Günü
eeprom 27
eeprom 28 // Isı Üst Limit Onlar
eeprom 29 // Isı Üst Limit Birler
eeprom 30 // Isı Alt Limit Onlar
eeprom 31 // Isı Alt Limit Birler
eeprom 32 // Isı Anlık Onlar
eeprom 33 // Isı Anlık Birler
eeprom 34 //
eeprom 35 //
eeprom 36 //
eeprom 37 //
eeprom 38 //
*/

```

## B.2. Basit Ses Kontrolleri

```

/*****
*
* Basit Ses kontrolleri
*
*****/

#define Buzz RA1 // Buzzer çıkışı
// tekrar sayısı,çalma süresi

```



```
extern void BSes(unsigned char,unsigned char,unsigned char);
```

### B.3. Delay

```
/*
 *   Delay functions for HI-TECH C on the PIC
 *
 *   Functions available:
 *       DelayUs(x)   Delay specified number of microseconds
 *       DelayMs(x)  Delay specified number of milliseconds
 *
 *   Note that there are range limits: x must not exceed 255 - for xtal
 *   frequencies > 12MHz the range for DelayUs is even smaller.
 *   To use DelayUs it is only necessary to include this file; to use
 *   DelayMs you must include delay.c in your project.
 *
 */
/*
 *   Set the crystal frequency in the CPP predefined symbols list in
 *   HPDPIC, or on the PICC command line, e.g.
 *   picc -DXTAL_FREQ=4MHZ
 *   or
 *   picc -DXTAL_FREQ=100KHZ
 *   Note that this is the crystal frequency, the CPU clock is
 *   divided by 4.
 *
 *   MAKE SURE this code is compiled with full optimization!!!
 */
#ifndef XTAL_FREQ
#define XTAL_FREQ 4MHZ          /* Crystal frequency in MHz */
#endif
#define MHZ *1000L              /* number of kHz in a MHz */
#define KHZ *1                  /* number of kHz in a kHz */
#if XTAL_FREQ >= 12MHZ
#define DelayUs(x)  { unsigned char _dcnt; \
```

```

        _dcnt = (x)*((XTAL_FREQ)/(12MHZ)); \
        while(--_dcnt != 0) \
            continue; }

#else

#define      DelayUs(x)  { unsigned char _dcnt; \
                        _dcnt = (x)/(((12MHZ)/(XTAL_FREQ))|1); \
                        while(--_dcnt != 0) \
                            continue; }

#endif

extern void DelayMs(unsigned char);

```

#### B.4. Dış Kontroller

```

/*****
*
*           Dış Kontroller           *
*
*****/

// EEprom 1 için 0 = Su Devirdaim Motoru kapalı 1 = Su Devirdaim Motoru açık
// EEprom 2 için 0 = Havalandırma Motoru kapalı 1 = Havalandırma Motoru açık
// EEprom 3 için 0 = Aydınlatma kapalı      1 = Aydınlatma açık
// EEprom 4 için 0 = Isıtıcı kapalı        1 = Isıtıcı açık
// DKon(parametre);
// 0 = Enerji Kesildi
// 1 = 1.Yemlik Motoru
// 2 = 2.Yemlik Motoru
// 3 = Aydınlatma
// 4 = Havalandırma Motoru
// 5 = Su Değiştir
// 6 = Isı değeri

extern void DKon(unsigned char);

#define YM1      RA1          // Yem1M
#define YM2      RA3          // Yem2M
#define EKes    RA5          // Şebeke Kontrol
#define HAVA     RB1          // Hava Motoru

```

```

#define AYDIN    RB2        // Aydınlatma
#define ISI      RB3        // Isıtıcı
#define ADSPRB4          // ( Akvaryum Dip Süpürme ) ADSPompaı
#define ATSVRB5          // ( Akvaryum Temiz Su ) ATSValfi
#define AKSV      RB6        // ( Akvaryum Kirli Su ) AKSValfi
#define ASev RC4          // Akvaryum Su Seviyesi

```

## B.5. DS1302

```

/*****/
/*          DS1302.H          */
/*****/

#define IO_TRIS TRISD5
#define IO      RD5
#define SCLK    RD4
#define RSTB    RD3
#define IO_TRIS TRISE1
#define IO      RE1
#define SCLKRE0
#define RSTB RE2

/*****/
/* Function Prototypes          */
/*****/

void  clkwritebyte(char ,char);
//void ramwritebyte(char ,char );
char  rbyte_3w();
void  reset_3w();
void  wbyte_3w(char);
void  initialize_DS1302();
//void burstramrd();
//void burstramwr(char);
//char read_clk_burst();
char  clkreadbyte(char);

```

## B.6. Ekran Tazeleme

```

/*****
*           Ekran Tazeleme           *
*****/

extern ETazele(void);

```

## B.7. Lcd

```

/*
*   LCD interface header file
*   See lcd.c for more info
*/

/* write a byte to the LCD in 4 bit mode */
extern void lcd_write(unsigned char);

/* Clear and home the LCD */
extern void lcd_clear(void);

/* write a string of characters to the LCD */
extern void lcd_puts(const char * s);

/* Go to the specified position */
extern void lcd_goto(unsigned char pos);

/* initialize the LCD - call before anything else */
extern void lcd_init(void);

extern void lcd_putch(char);

/*   Set the cursor position */
#define      lcd_cursor(x) lcd_write(((x)&0x7F)|0x80)

```

## B.8. LCD Mesajları

```

/*****
*           LCD Mesajları           *
*****/

// #define EKes      RA5           // Şebeke Kontrol

```

```

//#define HAVA    RB1                // Hava Motoru
//#define AYDIN   RB2                // Aydınlatma
#define ISI       RB3                // Isıtıcı
//      Seçilen Mesaj,LCD'ye Yazılacağı yer
//      yazılacak mesaj,Nereye yazılacak
extern int LCD(unsigned char,unsigned char);
extern int ZamanB(unsigned char);
extern int ZamanO(unsigned char);
//extern EKesildi(void);

```

### B.9. Uzaktan Kumanda kontrolleri

```

/*****
*      Uzaktan Kumanda kontrolleri      *
*****/

extern void U_Kum(void);

#define IR          RA2                // IR girişi

```

## ÖZGEÇMİŞ

Faruk AKYILDIZ, 1976 Sakarya-Hendek doğumludur. İlkokulu ve ortaokulu Sakarya-Adapazarı'nda, lise eğitimini Sakarya Fatih Anadolu Meslek Lisesi Elektronik bölümünde 1994 yılında tamamlamıştır. 1995 yılında girdiği Gazi Üniversitesi, Teknik Eğitim Fakültesi, Elektronik ve Bilgisayar Öğretmenliği bölümünden 1999 yılında mezun olmuş ve aynı yıl Elektronik öğretmeni olarak Sakarya Fatih Anadolu Teknik, Teknik ve Endüstri Meslek lisesine atanmıştır. 2005 yılından bu yana Sakarya Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Bölümünde Yüksek Lisans öğrencisidir.