

ARDUINO EĞİTİM KİTABI

GÖKHAN DÖKMETEŞ

İÇİNDEKİLER

BÖLÜM 1: ELEKTRONİĞE GİRİŞ	1
Analog Devre Elemanları ve Entegreler	2
Dirençler	3
Dirençlerde Renk Kodu ve Dirençleri Okumak	5
Kondansatörler (Kapasitörler)	6
Bobinler	9
Zener Diyot	10
Transistörler	11
Besleme ve Bataryalar	13
Güç Kaynağı	13
Pil	14
Elektronik Devreleri Anlama ve Devre Yapımı	14
Devre Kurma Yolları	15
BreadBoard	16
Elektronikte Kullanılan Araçlar ve Ölçme Aletleri	17
Lehim Teli	17
Breadboard Üzerine Devre Kurma ve Ölçüm Uygulamaları	20
Led Yakılım	20
555 ile Yanıp Sönen Işık	22
Çeşitli Elektronik Parçaları Ölçelim	25
Neler Öğrendik?	26
BÖLÜM 2: MİKRODENETLEYİCİLER	27
Mikrodenetleyiciler ve Mikroişlemciler	28
Mikrodenetleyici ve Mikroişlemci Arasındaki Farklar	28
Mikroişlemcilerin Yapısı ve Ayırıcı Özellikleri	29
Mikrodenetleyicilerin Belli Başlı Özellikleri	31

Günümüzde Üretilen Mikrodenetleyiciler ve Kullanım Alanları	34
PIC Mikrodenetleyiciler	34
ATMEL Mikrodenetleyiciler	35
STM32 Mikrodenetleyiciler	36
Mikrodenetleyici Benzeri Diğer Yapılar	36
PLC	36
SoC	37
Programlanabilir Lojik Diziler ve FPGA	37
Neler Öğrendik?	38
BÖLÜM 3: MİKRODENETLEYİCİ KARTLARI VE ARDUINO	39
Deneme Kartları ve Geliştirme Kartları	40
Yaygın Olarak Kullanan Mikrodenetleyici Kartları	41
ARM Geliştirme Kartları	41
Digispark Geliştirme Kartı	42
St Geliştirme Kartları	42
Amaca Özel üretilmiş kartlar	42
IOIO-OTG	44
Arduino	44
Arduino'da Kullanılan Mikrodenetleyiciler	45
Arduino Kartlar ve Özellikleri	46
Arduino UNO	46
Arduino Nano	48
Arduino Pro Mini	49
Arduino Mega	50
Arduino Leonardo	52
Arduino Pro Micro	53
Arduino Due	54
Diğer Arduino Modelleri	56

Amaca Uygun Olan Arduino'yu Seçmek	56
Program Hafızası	56
SRAM	56
EEPROM	57
Dijital Giriş Çıkış Ayakları	57
Analog Giriş Ayakları	57
İşlem Hızı	58
Fiyat	58
Hangi Kartı Seçmeliyim?	58
Klon Arduino Modelleri ve Önemli Hususlar	58
Klon Kartlar ve Driver Sorunu	59
Neler Öğrendik?	59
BÖLÜM 4: HAZIRLANMA VE GEREKLİ YAZILIMLAR	61
Arduino Dili ve Programlama Temelleri	62
Gerekli Araçların Temini ve Ortamı Hazır Hale Getirmek	65
Arduino Derleyicisini Kurmak	66
Klon Arduino için Sürücü Yükleme	67
Arduino Derleyicisinin Arayüzü	68
Menü Çubuğu Bileşenleri	68
Araç Çubuğu	72
Bilgi Çubuğu ve Bölmesi	72
Neler Öğrendik?	73
BÖLÜM 5: İLK PROGRAM VE BASİT GİRİŞ ÇIKIŞ	75
Arduino Uno Kartını Tanıyalım	76
RX0 ve TX0	76
5V ve 3V3	77
VIN ve GND	77
Analog Girişler	77

Karta İlk Programı Yükleme	78
void setup()	80
Tırnaklı parantez ve noktalı virgül	80
pinMode()	80
void loop()	81
digitalWrite()	81
delay()	81
Açıklama İşaretleri	82
Trafik Işığı Uygulaması	83
Düğmeleri Arduino ile Kullanmak	90
Arduino ile Nokia SMS Müziği	94
Özet	95
Neler Öğrendik?	96
BÖLÜM 6: DÖNGÜLER VE KARARLAR	99
İlişki Operatörleri	100
Döngüler	101
for	101
While	106
Serial.read()	109
do	111
Kararlar	112
if	112
else	116
switch	118
Mantık Operatörleri	121
Etiketler	123
Neler Öğrendik?	123

BÖLÜM 7: ARİTMETİK OPERATÖRLER VE MATEMATİK İŞLEMLERİ	125
Değişkenler	126
void	126
boolean	126
char	126
unsigned char	127
byte	127
int	127
unsigned int	127
word	128
long	128
unsigned long	128
short	128
float	129
double	129
const	129
Değişkenlerin Dönüşümü	129
Aritmetik Operatörler ve Dört İşlem	130
Birleşik Operatörler	132
Matematik Fonksiyonları	133
min()	133
max()	133
abs()	134
constrain()	134
map()	134
pow()	134
sqrt()	134
sin()	134

cos()	135
tan()	135
Bit Bazlı İşlemler	135
AND (&)	135
XOR (^)	136
OR ()	136
NOT (~)	137
Sola kaydırma (<<) ve Sağa kaydırma (>>)	137
Neler Öğrendik?	139

BÖLÜM 8: DİZİLER VE STRING **141**

Diziler	142
Karakter Dizileri	147
String	149
String Fonksiyonları	155
String()	155
compareTo()	156
concat()	157
c_str()	158
endsWith()	158
equals()	158
indexOf()	159
lastIndexOf()	159
length()	160
remove()	160
replace()	160
Neler Öğrendik?	161

BÖLÜM 9: FONKSİYONLAR	163
Basit Fonksiyonlar	164
Fonksiyona Argüman Aktarmak	168
Fonksiyondan Değer Döndürmek	169
Referans Argümanlar	171
Aşırı Yüklenen Fonksiyonlar ve Varsayılan Argümanlar	173
Neler Öğrendik?	175
BÖLÜM 10: ANALOG GİRİŞ VE ANALOG DİJİTAL ÇEVİRİCİ	177
Basit Analog Devreler	178
LM35 ile Dijital Termometre	183
Joystick Uygulaması	187
ADC Üzerinde İşlem Yapmak	189
Neler Öğrendik?	191
BÖLÜM 11: KARAKTER LCD EKРАН VE LIQUIDCRYSTAL KÜTÜPHANESİ	193
LCD Ekranın Ayakları ve Karta Bağlanması	196
LiquidCrystal Kütüphanesi	200
LiquidCrystal();	201
begin();	202
clear();	202
home();	203
write();	204
print();	204
cursor();	205
noCursor();	205
blink();	206
noBlink();	206
display();	206
noDisplay();	206

scrollDisplayLeft();	206
scrollDisplayRight();	206
autoscroll();	207
noAutoscroll();	207
leftToRight()	207
rightToLeft()	207
createChar()	207
Karakter LCD Seri iletişim (I2C) Modülü	209
Karakter LCD Uygulamaları	212
Neler Öğrendik?	221
BÖLÜM 12: PWM (PULSE-WIDTH MODULATION) VE TON UYGULAMALARI	223
PWM	224
PWM Uygulamaları	225
Tone Fonksiyonu	230
Neler Öğrendik?	233
BÖLÜM 13: DÂHİLİ EEPROM VE EEPROM İŞLEMLERİ	235
EEPROM Kütüphanesi	237
read()	237
update()	238
get()	238
write()	238
put()	239
EEPROM[]	239
Dâhili EEPROM ile Uygulamalar	240
LM35 Datalogger	240
EEPROM Okuma	247
Şifreli Kilit	248
Neler Öğrendik?	255

BÖLÜM 14: SPI VE I2C İLETİŞİM PROTOKOLLERİ	257
SPI (Serial Peripheral Interface)	258
SPI Kütüphanesi	264
SPISettings	264
begin()	266
end()	266
beginTransaction()	266
endTransaction()	266
transfer(), transfer16()	267
SPI Uygulamaları	267
SD Kart Kullanımı ve SD Kütüphanesi	275
SD Kütüphanesi	277
begin()	277
exists()	278
mkdir()	278
open()	278
remove()	279
rmdir()	279
available()	279
close()	279
flush()	280
peek()	280
position()	280
print()	280
println()	281
seek()	281
size()	281
read()	282

write()	282
isDirectory()	282
openNextFile()	282
rewindDirectory()	283
SD Kart Uygulamaları	283
I2C (Inter-Integrated Circuit)	292
Wire Kütüphanesi	297
begin()	297
requestFrom()	297
beginTransmission()	297
endTransmission()	298
write()	298
available()	298
onReceive()	299
onRequest()	299
I2C Uygulamaları	299
RTC (Real-Time Clock) ile Saat Uygulaması	300
Neler Öğrendik?	305
BÖLÜM 15: ARDUINO'YU İNTERNETE BAĞLAMAK	307
Ethernet Kalkanı	308
Ethernet Kütüphanesi	310
Ethernet Uygulamaları	318
Ethernet Modülü	324
Neler Öğrendik?	326

BÖLÜM 16: KABLOSUZ İLETİŞİM	329
nRF24L01+ Modülü	330
RF Alıcı ve Verici	336
RFID	341
Bluetooth	347
Bu Bölümde Ne Öğrendik?	351
BÖLÜM 17: GELİŞMİŞ GÖSTERGELER VE EKРАНLAR	353
LED Göstergeler	354
İki Renkli LED	354
RGB Led	356
7-Segment LED Gösterge	358
Dotmatrix Gösterge	368
Ekranlar	377
Nokia 5110 Ekranı	377
TFT Ekranlar	383
DS18B20 ile TFT Termometre ve One-Wire Protokolü	386
Neler Öğrendik?	390
BÖLÜM 18: DONANIM ÖZELLİKLERİNİ KULLANMA	393
Güç Tasarrufu	394
Kesmeler (Interrupt)	401
Zamanlayıcı (Timer)	404
Neler Öğrendik?	407
BÖLÜM 19: ARDUINO İLE MOTOR KONTROLÜ	409
Motorlar	410
Doğru Akım Motoru	410
Step Motor	411

Servo Motor	412
Motor Kontrolü	412
attach()	415
write()	416
writeMicroseconds()	416
attached()	416
detach()	416
Robot Araba Uygulaması	419
Neler Öğrendik?	426
BÖLÜM 20: PROJE ÜRETMEK	429
İnternette Bilgi Edinmek	435
Arduino Kütüphaneleri	437
Kalıcı Devre Tasarımı	438
Arduino Kalkanları	440
Arduino IDE Uyumlu Kartlar	441
Yazılımlar	441
Neler Öğrendik?	442
BÖLÜM 21: ARDUINO İLE PROJELER VE UYGULAMALAR	443
Dungeons & Dragons Zar Programı	444
Gelişmiş Robot Araba	452
TFT Seri Port Ekranı	455
İki Arduino Arasında UART İletişim	458
Arduino'yu Televizyona Bağlamak (TVOut)	461
Arduino'yu Monitöre Bağlamak	462
BMP180 ile Yükseklik Ölçer (Altimetre)	465
Arduino ile Yılan Oyunu	468

Arduino ile Ultrasonik Radar	471
Arduino ile Yağmur Algılama	474
5110 Ekranı ile Osiloskop Yapımı	475
Flappy Bird Oyunu	479
LM35 ile Analog Göstergeli Termometre	486
Ucuz Renk Algılayıcısı Yapımı	488
Arduino ile LED Zar	492
Web Sunucuya Sıcaklık ve Nem Değerlerini Yollama	495
Herhangi Bir Algılayıcı Olmadan Sıcaklık Ölçmek	498
RJ45 Kablo Sağlamlık Ölçer	499
ACS712 ile Akım Ölçme	502
PIR Hareket Algılayıcısı	503
Arduino Polis Flaşörü	505
RGB Led ile Renk Oyunu	507
Arduino ile RSS Besleme Göstergesi	511
Arduino'ya Linux, Wi-Fi, Ethernet ve USB Ekleme	516
Arduino ile Binary Saat	518
7 Segment Gösterge ile Termometre	522
Arduino Oyun Yayıncısı	523
Arduino ile Ateş Algılayıcı	525
Karakter LCD kullanarak Twitter Durum Ekranı	526
5110 Ekranı ile 2 kişilik Masa Tenisi Oyunu	532
Örnek Kodlar ve Kaynakça	540
Kaynakça	540
İndeks	541

1

ELEKTRONİĞE GİRİŞ

BU BÖLÜMDE

Analog Devre Elemanları ve Entegreler	2
Besleme ve Bataryalar	13
Elektronik Devreleri Anlama ve Devre Yapımı	14
Breadboard Üzerine Devre Kurma ve Ölçüm Uygulamaları	20
Neler Öğrendik?	26

Bu bölümde elektroniğin esaslarını ve devre elemanlarını görerek kitabımıza giriş yapacaksınız. Sonrasında Breadboard (Devre Tahtası) üzerine devre kurmayı göreceğiz. Bu iki husus da kitabın ilerleyen bölümlerindeki projeleri yapmanız için hatta kendi projelerinizi yapmanız için gereklidir. Bir sonraki bölümde ise elektronikte kullanılan araçlar ve ölçme aletlerine giriş yapacağız ve bunların nasıl kullanılması gerektiğini anlatacağız.

Kitabımızda bir çok modül ve kart kullanacağız ve uygulamalara çokça yer vereceğiz. Özellikle yeni başlayan okurların ileri seviyeye kadar gelmesini istediğimiz için, daha kitabın ilk bölümlerde elektronik hakkında yeterli bilgiye sahip olmanızı hedefliyoruz.

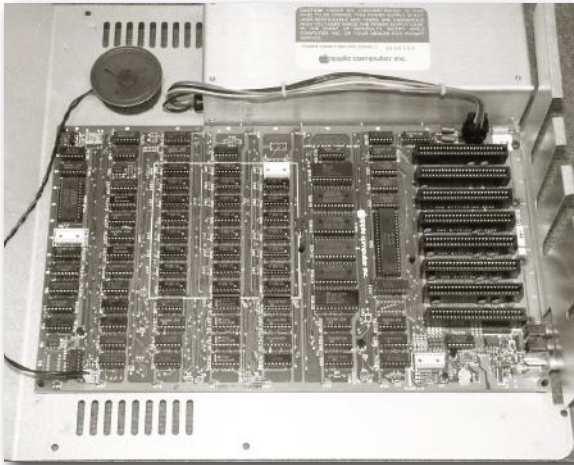
Elektronikten bahsetmemiz sizi endişelendirmesin. Çeşitli elektronik formülere, parçaların karakteristiklerine ve diğer hesaplamalara değinmeyeceğiz. Birçok hevesli kişiyi en başta elektronikten soğutacağı için aşırı derecede ayrıntılı teorik bilgi vermekten mümkün olduğunca kaçındık. Eğer hesaplamaya ihtiyacınız olursa internetteki çeşitli elektronik hesaplama programları oldukça işe yarayacaktır. Elektronikte pratiğin teoriden ileride olduğunu bundan yüz yıl önce yaşamış bilim insanları da fark etmiştir. Örneğin İtalyan Guglielmo Marconi "Teori ve formüller ortaya atmam. Pratikte ne yapılması gerektiğini fark ederim ve hemen hemen her zaman doğru yolu bulurum" diyerek kendi zamanına kadar telsiz tekniğinde ortaya koyulmuş bilgilerden faydalanarak mors işaretlerini kilometrelerce öteye ulaştırana kadar deneyler yapmıştır. O yüzden anlayarak pratik yapmak anlamadan sayfalarca teorik bilgi okumaktan daha faydalıdır.

ANALOG DEVRE ELEMANLARI VE ENTEGRELER

Bir çoğumuz elektrikle doğup büyüdüğü için elektriğe hiç yabancı değildir. O yüzden elektriğin ne olduğunu, nasıl çalıştığını, nasıl üretilip evlerimize geldiğini anlatma ihtiyacı duymuyoruz. Zaten elektronikte sizi elektrik prizinin ötesi pek alakadar etmez. Herhangi bir elektronik cihaz karşımıza kutulanmış ve kullanıma hazır bir şekilde sunulur. Bu kutunun üzerinde genellikle kullanıcının anlamakta zorlanmayacağı düğmeler ve anahtarlar olur.

Bir radyo cihazını ele alalım. Üzerinde sadece birkaç düğme ve döndüreceğimiz yerler vardır. Bu herkesçe bilinen ve çocuk oyuncuğu olan bir şeydir. Fakat elinize bir tornavida alıp radyonun içini açtığınızda çok farklı parçalar ile karşı karşı-

ya kalırsınız. Bu hiç radyonun dış kısmına benzemez ve her bir parçanın özel bir görevi vardır. Bunu bilmek için elektronikçi gördüğü devre kartındaki parçaları tanımalı, parçaların datasheet'lerini (veri dosyası) okumalı ve parçalar hakkında yeterli bilgiye sahip olmalıdır.



İlk kişisel bilgisayarlardan biri olan Apple II'nin anakartı. (resim:classicmp.org)

4

HAZIRLANMA VE GEREKLİ YAZILIMLAR

BU BÖLÜMDE

Arduino Dili ve Programlama Temelleri	62
Gerekli Araçların Temini ve Ortamı Hazır Hale Getirmek	65
Arduino Derleyicisini Kurmak	66
Arduino Derleyicisinin Arayüzü	68
Neler Öğrendik?	73

Arduino programlamaya başlamak için tek kalan şey artık bir Arduino kart edinmek ve gerekli yazılımları yüklemektir. Fakat bunun öncesinde Arduino dilinden ve bir programın nasıl çalıştığından kısaca bahsetmek istiyoruz. Sonrasında ise Arduino derleyici kurulumu ve kartın bağlanmasını açıklayacağız.

ARDUINO DİLİ VE PROGRAMLAMA TEMELLERİ

Mikrodenetleyicileri programlamak için çeşitli diller kullanılmaktadır. Bu diller Assembly, C gibi dillerdir. Bu iki dil mikrodenetleyicileri programlamak için kuvvetli dillerden ikisidir. Yalnız Assembly dili ilkel bir dil olduğu için anlaşılması zordur. Üstelik alt seviye bir dil olduğu için diğer üst seviye dillere göre oldukça uzun kod yazmak gerekmektedir. Assembly dilinde program yazmak konuşurken kelime kelime değil de kelimeleri harf ile kodlayarak konuşmaya benzer.

```
title Merhaba Dünya Programı

dosseg
.model small
.stack 100h

.data
hello_message db 'Merhaba Dünya!',0dh,0ah,'$'

.code
main proc
    mov ax,@data
    mov ds,ax

    mov ah,9
    mov dx,offset hello_message
    int 21h

    mov ax,4C00h
    int 21h
main endp
end main
```

Yukarıdaki kodda **Assembly** dilinde **Merhaba Dünya** yazdık. Aynı program **C** dilinde şöyle yazılabilir.

```
#include<stdio.h>

int main()
{
```

İPUÇU

Arduino programlamayı öğrenmek uygulamaya dayandığı için herhangi bir simülasyon veya kâğıt üzerinde öğrenmenizi tavsiye etmiyoruz. Kitabımızda da sıkıcı teorik anlatım ve hesaplamalara asgari düzeyde yer verdik. Onun için okuyucu gerekli araç ve gereçleri temin etmelidir.

ARDUINO DERLEYİCİSİNİ KURMAK

Arduino'nun resmi sitesi olan (Arduino.cc) 'de Arduino kartlar hakkında çeşitli donanım bilgilerini ve örnek kodları bulabilirsiniz. Arduino karta bilgisayardan program yüklemek ve çeşitli sürücülerin kurulması için Arduino IDE yazılımını yüklemek gerekmektedir. Bu yazılımı ücretsiz olarak resmi siteden indirebilirsiniz.

The screenshot shows the Arduino IDE download page. At the top, there is a large circular logo with a minus sign and a plus sign. To the right of the logo, the text reads: **ARDUINO 1.6.7**. Below this, it says: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the Getting Started page for Installation Instructions." To the right of this text, there are links for "Windows Installer", "Windows ZIP File for non-admin install", "Mac OS X 10.7 Lion or newer", "Linux 32 bits", and "Linux 64 bits". Below these links are "Release Notes", "Source Code", and "Checksums".

Below the main content, there are two sections: "ARDUINO SOFTWARE HOURLY BUILDS" and "ARDUINO 1.0.0 / 1.5.x / 1.6.x PREVIOUS RELEASES". The "Hourly Builds" section includes a "LAST UPDATE" box with the date "23 December 2015 17:15:19 GMT" and a description: "Download a preview of the incoming release with the most updated features and bugfixes." It lists supported operating systems: "Windows", "Mac OS X (Mac OS X Lion or later)", and "Linux 32 bit, Linux 64 bit". The "Previous Releases" section includes a description: "Download the previous version of the current release, the classic Arduino 1.0.x, or the Arduino 1.5.x Beta version." and a note: "All the Arduino 00xx versions are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X."

Tarayıcınızdan **Arduino.cc** sitesine girdikten sonra **Download** sekmesine tıklayın. Karşınıza yukarıdaki resimdeki gibi bir ekran çıkacaktır. Eğer Windows kullanıyorsanız **Windows Installer** kısmına tıklayarak kurulum dosyasını indirin. Linux veya Mac OS işletim sistemleri için de Arduino geliştirme ortamının sürümleri mevcuttur. Hangi işletim sistemini kullanıyorsanız ona uygun kurulum dosyasını indirmeye dikkat edin.

İndirme sayfasındaki **Hourly Builds** kısmı gelecek olan Arduino geliştirme ortamının en son halini içerir. Eğer yeni derleyiciye yeni özellikler eklenecekse buradan indirip bakabilirsiniz.

8

DİZİLER VE STRING

BU BÖLÜMDE

Diziler	142
Karakter Dizileri	147
String Fonksiyonları	155
Neler Öğrendik?	161

Bu bölümde dizi değişkenleri ve ardından bir karakter dizisi olarak çalışan string ifadeleri göreceğiz. String değişkenler metin tabanlı işlemlerde kullanılır. Integer gibi değişkenler içerisine sadece tamsayı değerleri veya sayı değerlerini alır. Bu değer de tek bir değer olabilir. Karakter dizileri bu eksikliği gidermektedir.

DİZİLER

Günlük hayatımızda benzer özelliğe sahip nesnelere belli bir yerde toplarız. Örneğin yiyecekler buzdolabında, para ve kartlar cüzdanda tutulur. Arduino programlamada da aynı özelliğe veya ortak noktaya sahip değişkenleri de bir noktada tutmak için diziler kullanılır. Diziler pek çok programlama dilinde kullanılmaktadır. Arduino programlarında kullanmak da oldukça kullanışlı olmaktadır. Diziler tanımlanırken değişkenleri tanımlamaya benzer bir yol izlenir.

```
int dizi [5];
```

Birinci deyim dizinin tipini belirler. Herhangi bir değişken tipinde dizi tanımlanabilir. İkinci deyim ise dizinin adını belirler ve köşeli parantez içindeki değer de dizinin büyüklüğüdür.

Buna benzer bir yapı tanımlamak için ayrı ayrı `dizi0`, `dizi1`, `dizi2`, `dizi3`, `dizi4` değişkenleri tanımlamamız gerekirdi. Ayrıca dizinin indeks numarası olan köşeli parantez içindeki rakam işlem yaparken dizi elemanlarına erişim kolaylığı sağlamaktadır. Ayrı ayrı tanımlanan değişkenlerde bu kolaylık yoktur.

<code>dizi0</code>	<code>dizi1</code>	<code>dizi2</code>	<code>dizi3</code>	<code>dizi4</code>
Değer	Değer	Değer	Değer	Değer

İPUCU

Arduino dilinde çoğu değer gibi dizinin indeks değerinin de 0'dan başladığını unutmayın. Dizinin indeks değeri 5 olarak belirtildiğinde bu değerler 0-4 arası olabilir.

Şimdi örnek programda bir dizi tanımlayıp bu elemanların hepsine erişip sonrasında da ekrana yazdıralım:

Kod

```
int dizi [5];
void setup() {
  Serial.begin(9600);
  dizi [0] = 55;
  dizi [1] = 65;
  dizi [2] = 10;
  dizi [3] = 58;
  dizi [4] = 2000;
```

```
for (int i=0 ; i<5 ; i++)
Serial.println(dizi[i]);
}

void loop() {
}
```

Program öncelikle dizi isminde 5 integer değişkeni boyutunda dizi tanımlar. Sonrasında dizi değerlerine tek tek değer verilir. Hangi dizi değerine değer vermek istiyorsak onun numarası köşeli parantez içine yazılır. Sonrasında bir for döngüsü ile bütün dizi değerlerini ekrana yazdırırız. Dizi değişkenlerine erişim için sayı yazılabildiği gibi değişken de yazılabilir. Değişkenin değerine göre dizi elemanına erişim sağlanır. `Serial.println(dizi[i])` fonksiyonunda da aynen bu yapılmıştır. Bu programla dizi değerlerini fonksiyonun içindeki paranteze yazabileceğimizi de anlıyoruz.

Dizilere ilk değeri programın içinde verebileceğimiz gibi dizileri ilk tanımladığımızda da verebiliriz. Bunun için şöyle bir kod yazmak lazımdır:

```
int dizi [5] = { 55, 65, 10, 58, 2000};
```

İlk değeri vermek için değerlerin tırnaklı parantez içine virgül ile ayrılarak yazıldığını unutmayın. Dizilere ilk değeri böyle vereceksek dizinin boyutunu yazmamıza gerek kalmaz. Derleyici bunu ilk verilen değerleri hesaba katarak bizim için yapar.

```
int dizi [] = {55, 65, 10, 58, 2000};
```

Böyle bir kullanım içerisinde çokça değer alan diziler için kullanışlı olabilir. Bu sayede ne eksik ne fazla dizi boyutu tanımlamış oluruz.

Dizileri tek boyutlu kullanabildiğimiz gibi çok boyutlu kullanmak da mümkündür. Çok boyutlu diziler kullanılırken bize daha dikkat etmek gerekmektedir. Şimdi çok boyutlu dizileri kullanan örnek bir program yazalım.

SPI VE I2C İLETİŞİM PROTOKOLLERİ

BU BÖLÜMDE

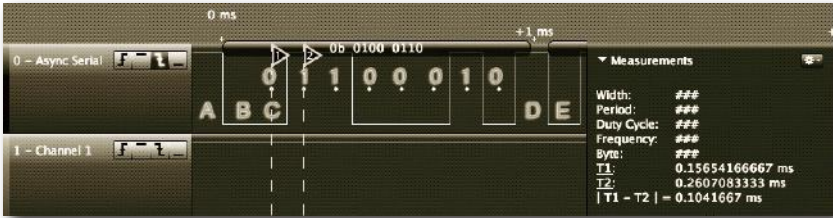
SPI (Serial Peripheral Interface)	258
SPI Kütüphanesi	264
SPI Uygulamaları	267
SD Kart Kullanımı ve SD Kütüphanesi	275
SD Kütüphanesi	277
SD Kart Uygulamaları	283
I2C (Inter-Integrated Circuit)	292
Wire Kütüphanesi	297
I2C Uygulamaları	299
RTC (Real-Time Clock) ile Saat Uygulaması	300
Neler Öğrendik?	305

SPI ve I2C hem Arduino kartlarda hem de mikrodenetleyici tabanlı uygulamalarda oldukça sık kullanılan iki protokoldür. Bu protokolle iletişim esasları belli bir standarta bağlanmış ve böylece birçok elektronik modül ve cihaz birbirleriyle aynı dilden konuşur olmuştur. Bu iletişim protokolleri bizim bilgisayarlarda kullandığımız USB, HDMI, ETHERNET gibi protokollere benzer. Bunları onlardan ayıran özellik daha basit olup alt seviye katmanda çalışmalarıdır. Bu bölümde bu iki iletişim protokolünü anlatacağız ve çeşitli modüller ile uygulama yapacağız.

SPI (SERIAL PERIPHERAL INTERFACE)

Tam Türkçe karşılığı **Seri Çevrelik Arayüz** olan bu iletişim birimi adından da anlaşılacağı üzere bir sistemin çevresindeki birimlerle iletişim kurmasını sağlar. Bu iletişim sistemi Motorola firması tarafından özellikle gömülü sistemler için geliştirilmiştir. SPI gibi başka iletişim protokolleri olsa da bu protokoller ya bir özel amaca hizmet etmektedir ya da bu protokolleri geliştiren firmalar geliştirdikleri protokolü tutunduramamıştır. O yüzden şu an kullandığımız mikrodenetleyicilerde SPI ve I2C'den başka sayılı iletişim protokolü ve yolu bulunmaktadır.

Arduino kartlardaki mikrodenetleyiciler kendi içinde **UART** (*Universal asynchronous receiver/transmitter*) yani evrensel asenkron alıcı ve verici bulundurmaktadır. Bu iletişim protokolünü seri port ekranını kullanırken kullanmış olduk. Bu iletişim protokolü çok basit bir sistem olup sadece bir adet gönderim bir adet de alım bacağına sahiptir. Bu bacaklar herhangi bir cihaza bağlanarak veri aktarımı yapabilir. RX ve TX bacakları UART yani bizim tabirimizle seri iletişim bacaklarıdır. Bu sistemin dezavantajlarından biri iki farklı hızda çalışan cihazların birbiriyle iletişim kurarken sorun yaşayabilmesidir. Örneğin seri port ekranını kullanırken `Serial.begin(9600)` ile **baud rate** yani bit oranı seçimi yapıyorduk. Bu saniyede gönderilen bir sayısını ifade eden birimler dizisidir. Burada dikkat edilmesi gereken nokta ise bilgisayardan da aynı **baud rate** seçiminin yapılmasıdır. Aksi halde ekranda bozuk ve anlamsız veriler görülecektir. Birbirinden farklı baud rate oranına sahip parçalar da birbiriyle iletişim kuramamaktadır.



Yukarıdaki resimde asenkron iletişime örnek bir lojik grafik gösterilmiştir. A konumunda herhangi bir veri aktarımı olmazken (TX 1 durumunda) B konumuna gelince başlangıç biti yollanmıştır. Başlangıç biri 0 olarak yollar ve belli bir süre karşı cihaza veri yollanacağını haber vermek için beklenir. C kısmında ise veri iletimi başlamış ve F karakterinin bitleri sıra ile yazılmaya başlanmıştır. F karakterinin bit değeri B0100110'dır. Görüldüğü gibi 1 yollanması için ayak 1 konumunda "Belli bir süre" kalmaktadır. 0 yollanması için ise 0 konumunda belli bir süre kalmaktadır. Bu kalınan süre baud rate oranına göre değişir. Bu iletişim

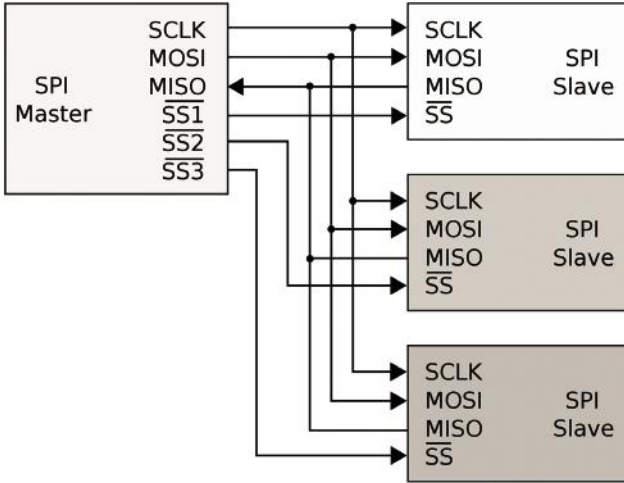
protokolünde verinin son biti ilk önce aktarılır. O yüzden ekranda 01000110 yerine 01100010 görürüz.

SPI ise eş zamanlı olarak çalışan bir veri yoludur. Yani hem veri sinyalleri gönderilirken bir yandan da saat sinyaliyle bunun zamanlaması yapılır. Böylece saat hızlarının farklılığından ortaya çıkan kopukluk ortadan kalkmış olur.

SPI iletişimde aygıtlar ana aygıt ve uydu aygıt olarak ikiye ayrılır. Ana aygıt uydu aygıtları kontrol edebilir ve bunlar üzerinde işlem yapar. SPI iletişim protokolünde ana aygıt için ayrı bir ayak ve uydu aygıt için ayrı bir veri ayağı kullanılmaktadır. **MOSI** (*Master Out Slave in*) ayağında ana aygıttan uydu aygıtlara giden veriler olur. **MISO** (*Master in Slave Out*) ayağında ise uydu aygıtlardan çıkan ve ana aygıta giden veriler iletilir.

SPI veri yolu üzerine yani 3 adet ayak üzerine sayısız aygıt bağlanabilir. Bu aygıtlardan birine veri göndermek istediğimizde istenilen aygıtın veri almasını belirlemek için **SS** veya **CS** (*Chip Select*) diye adlandırılan ayak kullanılır. Bu ayak bağlanan aygıt sayısına artar ve herhangi bir giriş çıkış birimi olabilir.

Özetlemek gerekirse SPI iletişim protokolünde **MOSI**, **MISO**, **SCK** ve **CS** diye adlandırılan 4 ayak kullanılmaktadır. Bu ayaklardan ilk üçü Arduino kartlarında belli noktalarda olup diğer **CS** ayağı kullanıcının seçimine bırakılmıştır.



Örnek bir SPI bağlantı şeması. Master yani ana cihaz birden çok uydu cihaza aynı hat üzerinde bağlanmış ve farklı CS ayakları üzerinden bağlanmıştır.

15

ARDUINO'YU İNTERNETE BAĞLAMAK

BU BÖLÜMDE

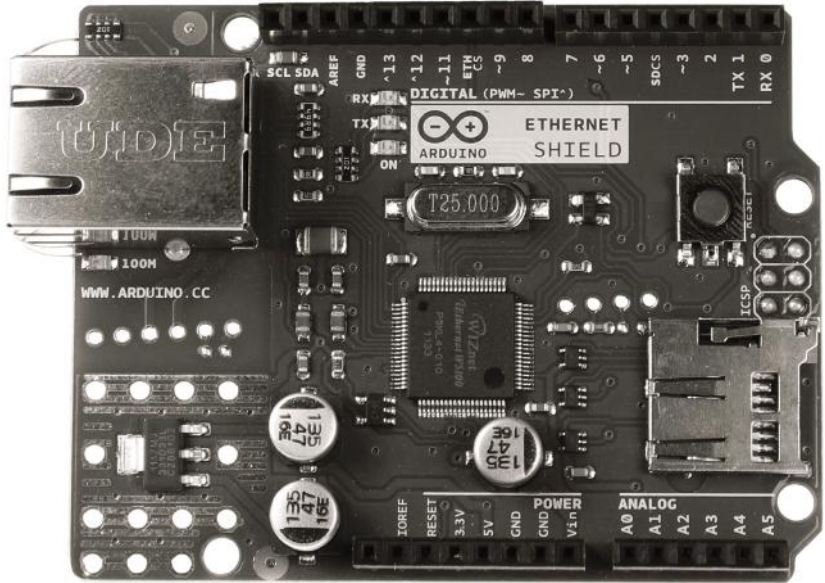
Ethernet Kalkanı	308
Ethernet Uygulamaları	318
Ethernet Modülü	324
Neler Öğrendik?	326

Arduino'yu bilgisayar gibi internete bağlamak mümkündür. Bunun için çeşitli genişleme kalkanları ve modüller vardır. Böylece kablolu veya kablosuz internete erişim sağlanmış olur. Bu bölümde internete bağlanma yollarını anlatacağız ve internet uygulamaları yapacağız.

Arduino'yu internete bağlamak pek çok yolla olabilir. Çeşitli firmalar mikrodenetleyicilerin internete bağlanması için çeşitli entegre ve modüller üretmiştir. Biz en kolay bulunabilir ve en ucuz modülleri yazımızda anlatacağız. Bu modüller ile Arduino aynı bir bilgisayar gibi internete bağlanabilir, internet sayfalarını açabilir, ağdaki diğer bilgisayarlarla iletişime girebilir veya sunucu olarak da kullanılabilir. Bunun için öncelikle modüllerin bağlantıları yapılmalı ve bu modüller için yazılmış kütüphaneler kullanılmalıdır. Şimdi Ethernet kalkanını tanıtmakla bölümümüze başlayalım.

ETHERNET KALKANI

Bu kalkan sayesinde Arduino'yu kısa bir zamanda internete bağlayabiliriz. Kart açık kaynak olduğu için de çeşitli klon modellerini Çin sitelerinden 5 dolara kadar bulabilirsiniz. Bu kartın internete bağlanması için **RJ45 Ethernet kablosu** gerekmektedir. Bu kablonun bir ucu kalkan üzerine bir ucu da evinizdeki veya çalışma alanınızdaki modeme bağlanmalıdır. Bu modül tek başına çalışmayıp en azından **Arduino Uno** modelinin üzerine takılması gereklidir. Eğer Mega kullanıyorsanız **Mega** ile de uyumludur.



Arduino Ethernet Kalkanı (resim:Arduino.cc)

DİKKAT

Her zaman kullanacağınız kalkanların hangi Arduino modeli ile uyumlu olduğunu kontrol edin. Bazı kalkanlar her ne kadar bazı modellere takılabilse de bunlarla uyum göstermeyebilir.

Eğer kartı herhangi bir dağıtıcı istasyona (HUB) takarsak normal bir kablo ile bağlantı yapabiliyoruz. Bilgisayara takılacaksa **cross-over** bağlantı şekliyle kablonun yapılması gerekiyor. Piyasada cross-over kabloyu bulmak da zor olduğundan kendi imkânlarınızla yapmanız gerekir.

Kartın üzerindeki WIZNet W5100 entegresi 10/100 Mbit hızında Ethernet kontrolcüsü olup mikrodenetleyicilerle SPI iletişim protokolü üzerinden bağlantı kurar. Bilgisayarın aksine internete bağlanmak için herhangi bir işletim sistemiyle kullanılması zorunlu değildir. Bu entegre aynı zamanda çeşitli cihazlarla beraber kullanılabilir. Ağ kameraları, Gömülü sunucular, tıbbi aygıtlar, otomasyon sistemleri, ağ yazıcıları ve çeşitli depolama aygıtları buna örnektir. Ayrıca bu entegre TCP/IP, TCP, UDP, IPv4, ICMP, ARP, IGMP ve PPPoE gibi çeşitli ağ protokollerini ve ilkelerini destekler.

Kartın üzerinde mikro SD kart yuvası da bulunmaktadır. Bu kart yuvası sayesinde SD kütüphanesiyle SD kart üzerinde çeşitli işlemler yapabiliriz.

Kart üzerindeki durum lambaları ise çeşitli durumları bize gösterir. Durum lambalarının bolluğu özellikle herhangi bir hata mesajını alamayacağımız durumlarda bize oldukça yardımcı olacaktır. Bu lambaların adı ve özellikleri aşağıdaki tabloda verilmiştir.

PWR	Karta güç geldiğinde yanar.
Link	Ağ linkini gösterir ve veri alışverişinde yanıp söner.
100M	100Mbit bağlantı olduğunda yanar.
FULLD	Ağ full duplex (Tam çift yönlü) olduğunda yanar.
COLL	IP çakışmalarında yanar.
RX	Veri alımında yanıp söner.
TX	Veri gönderiminde yanıp söner.

NOT

Karta ayrıca Arduino'nun 13. bacağına bağlı olan genel maksatlı LED de bulunmaktadır. Arduino Ethernet kartının şematiğini <https://www.arduino.cc/en/uploads/Main/arduino-ethernet-shield-06-schematic.pdf> bağlantısından inceleyebilirsiniz.