The Arduino is a cheap, flexible, open source microcontroller platform designed to make it easy for hobbyists to use electronics in homemade projects. With an almost unlimited range of input and output add-ons, sensors, indicators, displays, motors, and more, the Arduino offers you countless ways to create devices that interact with the world around you.

In Arduino Workshop, you’ll learn how these add-ons work and how to integrate them into your own projects. You’ll start off with an overview of the Arduino system but quickly move on to coverage of various electronic components and concepts. Hands-on projects throughout the book reinforce what you’ve learned and show you how to apply that knowledge. As your understanding grows, the projects increase in complexity and sophistication.

Among the book’s 65 projects are useful devices like:

- A digital thermometer that charts temperature changes on an LCD
- A GPS logger that records data from your travels, which can be displayed on Google Maps
- A handy tester that lets you check the voltage of any single-cell battery
- A keypad-controlled lock that requires a secret code to open
- An electronic version of the classic six-sided die
- A binary quiz game that challenges your number conversion skills
- A motorized remote control tank with collision detection to keep it from crashing

Arduino Workshop will teach you the tricks and design principles of a master craftsman. Whatever your skill level, you’ll have fun as you learn to harness the power of the Arduino for your own DIY projects.

John Boxall (http://www.tronixstuff.com/) has been writing Arduino tutorials, projects, and kit and accessory reviews for years. Arduino Workshop is his first book.
I N D E X

Symbols & Numbers
8, 139
&&, 73
*, 84
*/, 27
==, 71
!, 73
!», 71
/, 84
/*, 27
//, 27
>, 84
>>, 84
#define, 70
#include, 149
-, 83
c, 84
c=, 84
%, 133
+, 83
[], 139–140
||, 73
1N4004 diode, 51, 232
24LC512. See EEPROM
433 MHz receiver shield, 275
7-segment LED displays, 126–128
controlling with shift registers, 127–130
schematic symbol, 127
74HC595. See shift registers
7805 voltage regulator, 209
schematic symbol, 209

A
amperes, 35
analogRead(), 80
analogReference(), 85–86
analogWrite(), 48–49
and, 73
Arduino, 1
board types, 217–224
libraries. See libraries
microcontroller specifications, 217–218
ATmega2560, 218
ATmega328P-PU, 218
ATmega328P SMD, 218
SAM3X8E, 218
shields. See shields

sketches
adding comments to, 27
creating your first, 27
uploading, 30
verifying, 30
suppliers, 6
Arduino Due, 223–224
specifications, 218
Arduino LilyPad, 221–222
Arduino Mega2560, 222
specifications, 218
Arduino Nano, 221
Arduino Uno, 20
analog sockets, 22
boards compatible with, 219–220
DC socket, 20
digital I/O sockets, 22
onboard LED, 22
power connector, 20
power sockets, 22
reset button, 23
schematic symbol, 57
AREF pin. See reference voltage
arithmetic, 83–84
arrays
defining, 124
writing to and reading from, 125–126
ATmega2560 specifications, 218
ATmega328P-PU, 21, 210
Arduino equivalent pinouts, 213
pin labels, 214
schematic symbol, 210
specifications, 218
uploading sketches to, 214–217
microcontroller swap method, 214–215
using existing Arduino board, 215–216
using FTDI cable, 216–217
ATmega328 SMD, specifications, 218
attachInterrupt(), 185

B
battery tester, 80–83
BC548 transistor, 50
binary numbers, 116
displaying with LEDs, 119–121
game, 122–124
binary to base-10 conversion, 116–117
bitwise arithmetic, 139–141
   AND, 139
bitshift left and right, 140–141
   NOT, 140
   OR, 139–140
   XOR, 140
blinking
   an LED, 29–30
   a wave pattern, 43
Boarduino, 220
Boolean variables, 72
bootloader, 192
breadboard Arduino, 208–217
circuit schematic, 211
buttons. See push buttons
buzzerers. See piezoelectric buzzers
byte variables, 117

C

   capacitors, 60–62
      ceramic, 61
         schematic symbol, 61
electrolytic, 62
      schematic symbol, 62
measuring capacity of, 60–61
reading values of, 61
   cellular communications
controlling Arduino via text message, 360–363
making calls from Arduino, 356–358
SerialGSM library, 358
sending text messages from Arduino, 358–360
SM5100B GSM shield, 350
   changing operating frequency, 354–355
   configuring and testing, 352–353
      suitable antenna, 350
suitable power supply, 351
supported GSM network frequencies, 349
client.print(), 342
client.println(), 342
clock. See real-time clock
collision detection
   with infrared sensors, 249–251
   with ultrasonic sensors, 251–256
   comments, 27
   comparison operators, 72–73, 84
      and, 73
      not, 73
      or, 73
   compiling sketches, 30–31
   constants. See #define
crystal oscillators, 209–210
      schematic symbol, 210
current (electrical), 34

D

   Darlington transistors, 231–232
      schematic symbol, 231
data buses. See I2C bus; SPI bus
   #define, 70
delay(), 29
delayMicroseconds(), 252
dice, 113–115
digital clock. See real-time clock
digital inputs, 63
   activating, 70
   reading, 70
digital outputs
   activating, 28
   maximum current, 49
   pulse-width modulation, 48
digital rheostats, 318–320
digitalWrite(), 29
diodes, 50–51
   schematic symbol, 57
displaying binary numbers, 119
do while, 105–106
DS3232, 321. See also real-time clock

E

   EEPROM, 218, 309
      external, 309
      internal Arduino, 301–303
      Microchip 24LC512, 309
   EEPROM.h, 302
   EEPROM.read, 302
   EEPROM.write, 302
electricity
   current, 34
electricity power, 35
   voltage, 35
electronic components
   capacitors. See capacitors
crystal oscillators, 209–210
      schematic symbol, 210
   Darlington transistors, 231–232
      Schematic symbol, 231
digital rheostats, 318–320
diodes, 50–51
      schematic symbol, 57
DS3232, 321. See also real-time clock
   EEPROM. See EEPROM
infrared receivers, 286
   LCDs. See liquid crystal displays (LCDs)
   LEDs. See light-emitting diodes (LEDs)
microswitches, 243–246
      schematic symbol, 244
motors, 231–235
   demonstration circuit, 232–235
   stall current, 231
numeric keypad, 187–193
wiring to Arduino, 188
piezoelectric buzzers, 87
schematic symbol, 88
port expanders. See Microchip Technology MCP23017
potentiometer. See variable resistors
push buttons. See push buttons
relays, 51
schematic symbol, 58
resistors. See resistors
shift registers. See shift registers
serves. See servos
temperature sensors, 90
schematic symbol, 91
touchscreens. See touchscreens
transistors, 50
schematic symbol, 58
switching higher currents with, 50, 52
trimpots, 87
variable resistors. See variable resistors
Eleven, 219
else, 71
EtherMega, 222–223
Ethernet shield, 24, 328
EtherTen, 338

F
false (Boolean value), 72
farads, 61
flash memory, 218
float variables, 84
for, 47
Freeduino, 220
Freetronics
Eleven, 219
EtherMega, 222–223
EtherTen, 338
RTC module, 322
functions, creating your own, 95–98

G
GLCD.ClearScreen(), 155
GLCD.CursorTo(), 155
GLCD.DrawCircle(), 157
GLCD.DrawHoriLine(), 157
GLCD.DrawRect(), 157
GLCD.DrawRoundRect(), 157
GLCD.DrawVertLine(), 157
GLCD.FillRect(), 157
glcd.h. See graphic LCDs
GLCD.init(), 155
GLCD.PrintNumber(), 155
GLCD.Puts(), 155
GLCD.SelectFont(), 155

GLCD.SetDot(), 157
Global Positioning System. See GPS
(Global Positioning System)
GND. See ground
Google Maps
GPS (Global Positioning System)
Arduino shield, 258
displaying coordinates from, 261–263
displaying logged journeys on Google Maps, 268–269
displaying time from, 263–265
logging position data from, 265–267
receiver, 259
showing location on Google Maps, 263
testing GPS shield, 259–261
graphic LCDs. See also liquid crystal displays (LCDs)
Arduino library, 155
connecting to Arduino, 154
displaying graphics on, 157–160
displaying text on, 156
using with Arduino, 153–160
ground, 35
schematic symbol, 59

H
HD44780. See liquid crystal displays (LCDs)
higher-voltage circuits, 52–53
HTML, 342

I
I²C bus
Arduino connectors, 308
device address, 308
EEPROM. See EEPROM
port expanders. See Microchip Technology MCP23017
real-time clock. See real-time clock
receiving data, 309
transmitting data, 308
voltage warning, 308
IDE. See Integrated Development Environment (IDE)
if, 71
#include, 149
infrared distance sensor, 246–249
connection to Arduino, 247
detecting collisions with, 249–251
example sketch, 248
infrared remote control
controlling Arduino with, 289–290
example Sony codes, 288
receiver modules, 286
testing reception, 287–288
TSOP4138, 286
int, 46
to: integers, 46
Integrated Development Environment (IDE), 25
command area, 25
icons, 25, 26
installing
  on Mac OS X, 7–11
  on Ubuntu Linux 9.04 and later, 15–18
  on Windows 8, 7
  on Windows XP and later, 15
menu items, 25, 26
message window area, 25, 26–27
text area, 25, 26
title bar, 5
interrupts, 184–186
  configuring, 184
interrupts(), 185
ip(), 341
IP address, 341

K
keypad.h, 191
keypads. See numeric keypads
Knight Rider, 43
KS0066. See liquid crystal displays (LCDs)
KS0108B. See graphic LCDs; liquid crystal displays (LCDs)

L
lcd.begin(), 150
lcd.clear(), 152
lcd.createChar(), 150
lcd.print(), 150
LCDs. See liquid crystal displays (LCDs)
lcd.setCursor(), 150
lcd.write(), 153
LEDs. See light-emitting diodes (LEDs)
libraries, 169–173
  installing
    in Mac OS X, 170–171
    in Ubuntu Linux, 172–173
    in Windows XP and later, 171–172
light-emitting diodes (LEDs), 39–40
  Arduino onboard, 22
    calculating current flow, 40
    changing brightness with PWM, 47–49
  matrix modules, 135–137
    animation with, 145–146
    schematic symbols, 136
    using with Arduino, 141–146
  schematic symbol, 57
seven-segment, 126–128
  controlling with shift registers, 127–130
  schematic symbol, 127
  use of, 39
LilyPad Arduino, 221–222
linear, 86
liquid crystal displays (LCDs)
  Arduino library, 150
  character displays, 148
  creating custom characters, 152–153
  graphic. See graphic LCDs
  using with Arduino, 149–151
LiquidCrystal LCD(), 150
logarithmic, 86
long variables, 107, 180
loop(), 28

M
MAC address, 342
Maxim DS3232. See real-time clocks
Microchip Technology MCP23017, 313–315
Microchip Technology MCP4162, 318–320
micros(), 179–181
MicroSD memory cards, 173–176
  writing data to, 175–176
microswitches, 243–246
  schematic symbol, 244
millis(), 179–181
MISO pin. See SPI bus
modulo, 133
MOSI pin. See SPI bus
motors, 231–235
  demonstration circuit, 232–235
  stall current, 231
motor shield, 238–240
  connections, 239
multimeters, 38
  multiplying numbers, 106–107

N
noInterrupts(), 185
not, 73, 140
numeric keypads, 187–193
  wiring to Arduino, 188

O
ohms, 36
Ohm’s Law, 40
or, 73, 139–140
oscilloscopes, 64, 79–80
Parallax Ping)). See ultrasonic distance sensors
piezoelectric buzzers, 87
schematic symbol, 88
PIN. See numeric keypads
pinMode(), 28
planning your projects, 34
Pololu RP5 tank chassis, 235–242
controlling with infrared remote, 291–293
port expanders. See Microchip Technology MCP23017
potentiometers, 86–87
power (electrical), 35
PowerSwitch Tail, 93
ProtoScrewShield, 327
pull-down resistors, 65
pulseDuration(), 252
pulse-width modulation (PWM), 47–48
demonstrating, 49
output pins, 48
using with analogWrite(), 48–49
push buttons, 63
de-bouncing circuit, 66
pin alignment of, 63
schematic symbol, 63
simple example, 65–70
PWM. See pulse-width modulation (PWM)
radio-frequency identification (RFID)
125 kHz RFID reader, 296
controlling Arduino with, 299–301
defined, 295
reading tags, 297–298
time-clock system, 330–336
random(), 112
random numbers, 112
randomSeed(), 112
real-time clock
connecting to Arduino, 322
displaying time with, 322–326
reading the time from, 326
setting the time, 325
reference voltage, 84–86
external, 85
internal, 86
relays, 51
schematic symbol, 58
repeating functions, 46
resistors, 35–37
color bands, 36
power rating, 38
pull-down, 65
reading resistor values, 36
schematic symbol, 57
variable, 86–87
RF data modules
Arduino library, 272
demonstrating, 273–274
wireless remote control with, 272–277
RFID. See radio-frequency identification (RFID)
RTC module, 322
safety warning, 18
SAM3X8E, specifications, 218
schematic diagrams, 56
SCK pin. See SPI bus
SCL pin. See I²C bus
SDA pin. See I²C bus
Serial.available(), 107
Serial.begin(), 102
serial buffer, 106
Serial.flush(), 107
SerialGSM.h, 358
serial monitor, 102–103
debugging with, 105
displaying text and data in, 102–103
sending data from serial monitor to
Arduino, 106–107
Serial.print(), 102
Serial.println(), 103
servo, 228
.attach(), 228
.write(), 228
servo.h, 227
servos, 226–230
connecting to an Arduino, 227
demonstrating, 227–228
required Arduino functions, 227–228
selecting an appropriate, 226
setup(), 28
shields, 23
Ethernet, 24, 328
microSD card, 163
numeric display/temperature, 24
prototyping, 164
creating your own, 165–169
shiftOut(), 121
shift registers
74HC595, 118
clock, 118–119
data, 118–119
latch, 118–119
schematic symbol, 119
sketches
adding comments to, 27
creating your first, 27
uploading, 30
verifying, 30
SoftwareSerial.h, 353
soldering, 167
soldering iron, 167
solderless breadboard, 41
SPI.begin(), 317
SPI bus, 315
  Arduino connectors, 316
digital rheostat, 318
receiving data, 309
transmitting data, 308
typical device connection, 316, 317
voltage warning, 308
SPI.h, 316
SPI.setBitOrder(), 317
SPI.transfer(), 317
SS pin. See SPI bus
stopwatch, 181–183
suppliers, 6
switch bounce, 64
switch... case, 190

T
tank chassis See Pololu RP5 tank chassis
tank robot, 235–242
Teleduino, 344
  blink codes, 347
  controlling Arduino with, 345–348
default pin status setup, 348
key, 345
messages from, 347
registering with, 345
temperature logging, 177–179
temperature sensor, 90
terminal emulator software, 280
then, 71
thermometer
  analog, 228–230
digital, 134–135
monitor, 157–160
quick-read, 90–92, 99–101
time. See real-time clock
timing with Arduino, 179–183, 321
TIP120, 231–232
  schematic symbol, 231
TMP36, 90. See also thermometer
  schematic symbol, 91
TO-220, 209, 232
touchscreens, 196–205
  breakout board, 196
  connections to Arduino, 196
  controlling Arduino with, 200
  mapping touchscreen area, 199
  traffic light simulator, 74–79
transistors, 50
  schematic symbol, 58
  switching higher currents with, 50, 52

U
ultrasonic distance sensors, 251–256
  connection to Arduino, 252
  detecting collisions with, 254
  example sketch, 252–253
unsigned long variable, 180
uploading sketches
  with FTDI cable, 216–217
  with the IDE, 31

V
variable resistors, 86–87
variables, 45–46
  byte, 117
  float, 84
  integer, 46
  long, 107, 180
verifying sketches, 30–31
void loop(), 28–29
void setup(), 28
virtual serial port, 353
virtualwire.h, 274
voltage (electrical), 35
volts, 35

W
watts, 35
web server, 339
while, 105
Wire.begin(), 308
Wire.beginTransmission(), 309
Wire.endTransmission(), 309
wire.h, 308
wireless data. See RF data modules; XBee
Wire.read(), 309
Wire.requestFrom(), 309
Wire.write(), 308

X
XBee, 277
  Arduino shield, 278
  explorer board, 278
  transmitting data with, 279–280
  using a remote control with, 281–283
XOR, 140