Index

A

Access Point Name (APN)
   AT+CGDATA command, 596
   AT+CGDCONT? command, 595
   AT+COPS command, 596
   AT&T 4G data connection, 594
   CGDCONT AT command, 595
   configuration, 593
Analogic keypad, 495
Analog I/O
   analogWrite–PWM, 114
   int analogRead, 114
   analogRead() function, 419
   Arduino reference (see Tone API)
DHT sensor library
   bits[] array, 211
   bytes value, 197
   code implementation, 202
   data readings, 204
   debugging, 213
   definition, 195
   DHT_4_Galileo class, 212
   DHT11 and DHT.cpp file, 202
   fastGpioDigitalRead() method, 210
   fastGpioDigitalWrite() method, 210
   Intel Galileo’s digital I/O headers, 197
   materials list, 195
   p_gate, 204
   pinMode() function, 197–198, 210
   p_read, 204
   “read” the sensor’s values, 197
   read() method, 204
   sendCommand() method, 204, 210
   single-wire two-way, 195
   temperature and humidity, 196
   tri-state buffers (see Tri-state buffers)
I/O speed
   digitalRead() function, 162, 164
   digitalWrite() function, 162, 164
   distribution, 165
   fastGpioDigitalRead()
      method, 173–175
   fastGpioDigitalRegSnapshot()
      method, 177, 179–180
   fastGpioDigitalRegWriteUnsafe
      method, 172, 177, 179–182
   fastGpioDigitalWrite()
      method, 171–172, 180
   fast I/O macros, 169
   INPUT_FAST, 166
   limitations, 163
   memory-mapped interface, 164
   north- and south-cluster pins, 182
   OUTPUT_FAST, 166
   pinMode() function, 162
   pinMode() method, 184
   port-mapped interface, 164
   setup() function, 162
pulseIn API
   code implementation, 188
   digital I/O header, 189
   IDE serial console, 190
   implementation, 189
   limitations, 189
   material, 189
   pinMode(), 190
   PWM, 188
INDEX

API (cont.)
  serial ports
    code implementation, 158
    limitation, 157
    materials list, 158
    println() method, 160–161
    schematic diagram, 158
    testing, 157
  servo motors (see Servo motors)
  Arduino integrated development environment (Arduino IDE)
    board connection, 96
    board selection, 103
  button
    code, 120
    materials list, 119
    schematics, 119
    sketch, 122
  debounce
    code, 125
    materials list, 125
    schematics, 125
    sketch, 127
  driver installation
    Linux, 100
    MacOS, 101
    Windows, 99
  drivers troubleshooting, 130
    oracle virtual box, 136
    serial communication issues, 130
    64-bit Linux,
      warning messages, 135
    VMware problems, 135
  fade
    code, 118
    materials list, 117
    schematics, 117
    sketch, 118
  firmware update process, 127
  Getting Start Guide, 94
  graphical interface, 102
  history, 93
  installation process
    Linux 32/64 bits, 95
    Linux machines, 98
    MacOS, 99
    MacOSX, 95
    Windows, 95, 97
  language reference and APIs
    analog I/O (see Analog I/O)
    digital I/O (see Digital I/O)
    loop() function, 113
    setup() function, 113
    time duration, 115
    makers community, 94
    port selection, 104
    ReadAnalogVoltage
      code, 123
      materials list, 122
      schematics, 122
      sketch, 124
    serial console, debugging
      (see Serial communication)
    7-zip tools, 96
    Sketch (see Sketch)
    sudo apt-get install
      p7zip-full command, 96
  Arduino reference page, 112
  AT+CGDCONT command, 596
  attach() method, 152
  available() method, 160–161

B

  Booting from SD card images
    card reader, 60
    copying files, 63
    Mac OS X, 61
    Ubuntu, 62
    USB adaptor, 60
    Windows, 61
  Booting from SPI card images, 64

C

  checkButtonState() function, 414, 419
  Consumer key, 311
  Consumer secret, 311
  Cypress CY8C9540A datasheet, 150, 191

D

  DC/DC converter, 500
  DediProg SF100, 86
  delay() function, 106
  Digital I/O
    digitalWrite, 113
    int digitalRead, 113
    pinMode, 113
    digitalRead() function, 119
    digitalWrite() function, 106, 119, 419
    drawMatrix() function, 414
Edge detection
- apertureSize argument, 363
- Canny function, 362
- cvtColor() function, 362
- opencv_capimage_canny.cpp, 361–363
- emailCounter.py script, 138

Emotions classification
- database creation, 373
  - CSV File, 380–382
  - directory, 379–380
  - image cropping, 374
  - pixel coordinates, 374
- faceDetect() method, 391
- FaceRecognizer code, 383
- fisherface model, 383
- happy face detection, 398
- image cropping, 393–395, 397
- image extraction, 398
- main() function, 391
- opencv_emotion_classification.cpp, 383–388
  - original code, 372
  - prediction method, 392
  - public algorithm, 392
  - surprised face detection, 399
  - testSample image, 392
- uvcvideo driver, 397
- void FaceRecognizer method, 392

Face and eyes detection
- CascadeClassifier() object, 364, 367
  - code’s sequence, 370–371
  - components, 367
- detectMultiScale() method, 368–370
- haarcascade_eye.xml code, 364
- haarcascade_frontalface_alt.xml code, 364
- opencv_face_and_eyes_detection.cpp, 365–367, 372
- Point center object, 370
- rectangle() and circle() functions, 364
- fastGpioDigitalRegSnapshot() method, 177
- fastGpioDigitalRegWriteUnsafe() method, 177, 179

Flashing, capsule files
- with Firmware Update Tool, 81
- with Linux Terminal Shell, 78
- with the Intel Arduino IDE, 77
- with UEFI Shell, 79

Flower face test, 410, 415

GET and POST methods, 478

Graphics Processing Units (GPUs), 324

Ground coffee gripper
  - air pump, 572
  - assembling, 573
  - control, 572
  - materials, 571
  - RoboticArm.ino, 575
  - RoboticArmWithCoffeeGripper.ino, 576
  - working principle, 570

Hacks
- library, 190
  - PWM
    - beginTransmission() method, 192
    - clock source, 191
    - custom_pwm.ino, 193
    - duty cycle, 191
    - endTransmission() method, 192
    - frequency, 191
    - implementation, 194
    - minimum granularity, 192
    - PLATFORM_NAME directive, 194
    - register period, 191
    - write() method, 192
- Home automation system
  - adding username and password, 497
  - analogic keypad, 495
  - cheerio REPL, 490
  - DHT11 sensor, 497
  - Ethernet cable, 425
  - home.html page, 491
  - LTE modem, 425
  - materials list, 428
  - PIR sensor
    - connection, 440
    - headers, 439
Home automation system (cont.)
9V battery, 441
software writing and testing, 442
PoE, 489
running, 486
SD card image, 425
sketch creation
join all test code, 455
network connections, 451
receiving UDP messages, 452
sending UDP messages, 451
software and hardware components
code review, 435
keypad code run, 437
keypad connection, 430
keypad functionality, 429
keypad software, 431
keypad testing, 428
software architecture
communication, 427
send commands, 427
UDP server listening, 427
web server, 426
TMP36 temperature sensor
connection with Intel Galileo, 448
software writing and testing, 450
voltage output, 448
web server, 426
web server creation
(see Web server creation)
WiFi module, 425
YwRobot relay module
connection with
External Lamps, 446
connection with Intel Galileo, 445
Intel Arduino digital ports, 443
LED and headers, 444
PNP 8550 transistor, 444
software writing and testing, 446
two-channel relay module, 443

I, J, K
IEEE 802.3af standard, 499
Intel Galileo boards
Arduino headers
analog ports, 12
analogReference() function, 12
I2C/two-wire interface, 12
input/output pins, 12
power, 12–13
source and sink currents, 13
average costs, 5
cables
serial debugging, 24–26
testing, 33
types, 23–24
components, 4, 9, 11
DipTrace tool, 3
Fritizing tool, 3
guitar model, 1–2
I2C address jumper, 15
Intel Galileo Gen 2
block diagram, 22
components, 18
Cypress GPIO expander, 18
elements, 23
FTDI terminals, 19
GPIO mapping, 23
issues, 22–23
JTAG connector, 20–21
jumpers and buttons, 21
MAX 3232, 19
power over ethernet, 19
serial cables, 26–28
user guide, 17
Intel Quark SoC X1000
code-named Clanton, 6
core processor, 7
Pentium opcode, 6
peripheral support, 7
security, 8
software support, 8
specifications, 7
inventors, 1
IOREF jumper, 14
limitations, 8
Maker Fair Rome, 2
mini-PCIe connector, 11
reboot button, 15
requirements, 3
reset button, 15
schematics
block diagram, 15
Intel Quark support, 16
multiplexers, 16
serial debugging, 16
voltage-level translator, 17
sketches, 4
VIN jumper, 14
Windows
busybox software, 32
COM port, 29
configuration, 30
hexadecimal sequence, 32
Mac OSX, 31
Ubuntu terminal, 30–31
Yocto build system, 2
Intel Galileo images
booting (see Booting from SD card images; Booting from SPI card images)
capsule files
compiling steps, 66
compiling UEFI firmware, 65
flashing (see Flushing, capsule files)
preparing environment, 65
preparing layout.conf, 70
troubleshooting, 69
using SPI tool, 77
cross-compiler toolchain
architectures, 52
Hello World program, 57
installation, 56
Linux, 54
OSX, 54
output files, 56
Windows, 55
debugging, 59
metafiles preparation, 47
preparing computer, 44
SPI Images Flash files (see SPI Images Flash files)
SPI vs. SD card images, 46
testing, 90

LAN IN port, 505
Leonardo da Vinci’s robot, 509
linux-cdc-acm.inf, 100
Logitech webcam C270, 321
loop() function, 105, 142, 155
LTE modem
antennas, 582
hardware settings
adapting modem card, 586
antennas connection, 587
Intel Galileo connection, 588
NGFF/mPCIe adaptor, 585
SIM card preparation, 583
internet bandwidth, 600
materials list, 581
project details, 581
project settings
CDC-ACM DRIVER loading, 603
Chat Script, 601
software settings
APN (see Access Point Name (APN))
CDC-ACM driver, 590
commands and responses, 591
internet connection, 600
IP Interface (see Point to Point Protocol Daemon (PPPD))
modem checking, 589
options-att configuration, 597–598
SIM card connection, 592
XMM 7160 and XMM 7260 modems, 580

Memory mapped buffers
(mmap) function, 333
Modem manager, 98
Motion JPEG encode, 330
myservo2.attach(3, true) method, 155

Networking and hacks
Ethernet API
DHCP connection, 247, 254
Ethernet.begin() method, 253
Ethernet class, 248
EthernetClient, 247
Ethernet objects removing, 271
EthernetUDP, 247
IPAddress and Server, 247
mac[] array, 248
NTP, 248
Quark SoC, 247
static IPs (see Static IPs)
WiFiUdpNtpClient.ino review, 252
sketch transfer
clloader application, 277
cupload_linux_ and_osx_hacked.sh, 284
Networking and hacks (cont.)
  clupload_linux.sh, 281
  clupload_osx.sh, 281
  clupload_win_hacked.sh, 284
  clupload_win.sh, 281
  code/hacked_platforms_files, 282
  configuration, 278
  file transfer mechanism, 277
  hacked IDE running, 285
  platform file, 282
  platform.linux64.txt, 281
  platform.win.txt file, 280
  SCP protocol, 279
  script reviewing, 285
  TCP/IP, 277
  tools.izmirld.cmd.path field, 281
  tools.izmirld.upload.
    pattern field, 281
  ZMODEM protocol, 276
transfer files
  ftp deamon, 272
  pscp.exe, 273
  scp tool, 273
  SD card, 273
  USB pen drive, 274
WiFi API
  ConnectWithWEP.ino, 233
  ConnectWithWPA.ino, 230
  firmware version, 228
  IDE serial console, 236
  Linux terminal shell, 237
  listNetworks() function, 228
  network’s key and key index, 236
  printEncryptionType(), 229
  printMacAddress(), 228
  ScanNetworks.ino, 225
  WiFi router, 237
  WiFi.begin(), 238
  WiFi.encryptionType(), 229
  WiFi.SSID(), WiFi.RSSI(), 229
  WL_NO_SHIELD, 228
WiFi cards
  antennas connection, 220
  bracket connection, 219
  BSP SD card image, 221
  iwlwifi driver, 218
  Linux terminal shell, 222
  materials, 218
  micro SD card, 221
  mini-PCle form factor, 218
  SD image, 218
  SD image upgradation, 223
  WiFi card and
    bracket connection, 220
WiFi library
  begin() methods, 243, 245
  connection restart, 242
  getLocalIP(), 245, 247
  int WiFiClass, 245–246
  iwconfig command tool, 245
  WEP connection, 241
  wlan0, 239
  WPA connection, 240
  wpapassphrase command, 245
Network Time Protocol (NTP), 248
Node package manager (npm), 467

O
OAuth secret, 311
OAuth token, 311
Open source Computer Vision (OpenCV)
  BSP image and toolchain
    build process, 325
    GPUs, 324
    rootfs size increment, 324
    Video4 Linux, eGlibc, 324
    Yocto project, 323
  command line, 353
  computer vision, definition, 319
  C++ vs. Python performance, 352
  development library package, 325
  device communication, 319
  fisherface model, 400
image capture
  buffer allocation, 350
  file transfer, 351
  flowchart, 353–354
  fprintf() function, 349
  magic identifier, 349
  mmap() function, 333
  opencv_capimage.cpp, 355, 358
  pixel format selection, 348
  PPM file extension, 349
  read/write application, 333
  RGB24 and BGR24, 348
  RGB24 format,
    352x288 resolution, 351
  software, requirements, 347–348
  storage, JPEG file, 355
userspace pointers, 333
V4L2, 353
YUYV to RGB24 conversion, 350
yuyv_to_rgb24() function, 350
y/-yuyv argument, 352
image processing
edge detection
(see Edge detection)
emotions
(see Emotions classification)
face and eye
(see Face and eyes detection)

Intel Galileo, 320
libraries, 320
materials list, 321
Python
opencv_capimage.py, 359
performance analysis, 360
robotic head, 400
USB video class (UVC), 322
V4L2, 320
video capture
buffer allocation, 337–338
buffer dequeue, 339–340
buffer enqueues, 338
-c argument, 343
device closed, 341
device initiation, 335
-f argument, 343
ffmpeg installation, video
conversion, 344
force_format variable,
resolution, 336
free() function, 341
image cropping, 336
IOCTL calls, 333
isOpened() method, 356
Mat class, 357
Mat object, 357
mmap() function, 333
MP4 file, 346
munmap() function, 341
-o argument, 343
open the device, 335
params argument, 358
properties, 356
querying process, 338
read/write application, 333
release() method, 357
sequence flowchart, 334
simple argument, 343
streaming process, 339, 341
userspace pointers, 333
VideoCapture class, 356
VIDIOC_G_FMT, 336
webcam capabilities, v4l2-ctl
camera properties, 329
Logitech C270, 327–328
pixel format and resolution, 330
set/change properties, 327
webcam connection, 326

P, Q

pinMode() function, 116, 119, 166
PoE injector TP-LINK TL-POE150S, 504
PoE system, 506
Point to Point Protocol Daemon (PPPD)
isp_chat script, 598
mknod command, 599–600
nodetach, 600
options-att configuration, 596–597
SPI and BSP images, 596
POWER+DATA OUT port, 505
Power of Ethernet (PoE), 489
Power over Ethernet (PoE)
advantages, 499
assembling, 500
connection, 505
goal, 499
LAN IN port, 505
materials, 500
POWER+DATA OUT port, 505
soldering, 506
TP-LINK TL-POE150S injector, 504
pulseln() method, 188
PythonP Code, 138

R

read() method, 160–161
Real time clock (RTC), 3
Robotic arm
actuators, 514
assembling
connecting U-shaped base, 522
elbow, 528
gripper, 535
preparing servos, 521
screws, 520

649
Robotic arm (cont.)
    shoulder, 526
    wooden base, 535
    wrist, 532
controlling (see Servo control board)
degrees of freedom, 510
materials, 519
online torque calculator, 515
parts and components, 516
perpendicular length, 512–513
stall torque units, 511–512

sendCommand() and read() methods, 204
Serial communication
    print messages, 109
    Serial.available() function, 109
    Serial.begin(int speed), 108
    serial monitor console, 110
    Serial.print(data) argument, 108
    Serial.read(), 109
setup() function, 111
Servo control board
    assembling, 544
    circuit Protection, 543
    external Power Supply, 543
    hardware, 538
    power supply, 537
    quantity of wires, 538
    servos control, 537
    software, 547
Thumbstick API
    challenges, 547
    header file, 548
    installation, 561
    modes, 548
    review, 558
    RoboticArm.ino, 561
    thumbsticks, 542
Servo motors
    code implementation, 148
    CY8C9540A, 150
    loop() function, 155
    materials list, 147
    MIN_PULSE_WIDTH and MAX_PULSE_WIDTH methods, 150
    myservo2.attach(3, true) method, 155
    schematic diagram, 147, 153
setup() function, 155
specifications, 155
testing, 154
theory, 145
uint8_t Servo::attach(int pin,
    int min, int max, bool
    force48hz = false), 152
uint8_t Servo::attach(int16_t pin,
    bool force48hz = false), 152
void Servo::set48hz(), 152
void Servo::set188hz(), 152
write() method, 149
writeMicroseconds() method, 150
serial monitor console, 110
review code
checkButtonState() function, 414
drawMatrix() function, 414
loop() function, 414
setup() function, 413
tweeting, 424
SPI Images flash files
binary file, 85
Ethernet MAC address, 82
flashing, 86
sample-platform-data.ini, 83
Static IPs
configuration, 268
MacOSX 10.0
DNS servers, 266
Ethernet cable
connection, 266–267
internet connection
sharing, 267–268
IP configuration, 264–265
Thunderbolt to gigabit
Ethernet adapter, 264
Ubuntu 12.04
IPv4 properties, 262
IPv4 settings, 261
network connections, 260
wired connection, 260
Windows 7
internet connection sharing, 258
ipconfig command, 257
IPv4 properties, 256–257
local area connection
properties, 255–256
new IP, 260
Wireless Lan Adapter, 258
wireless adapter, 255
sudo apt-get install
p7zip-full command, 96

T
Thumbstick API
challenges, 547
RoboticArm.ino
review, 566
running, 568
sketch for Intel Galileo boards, 561
Tone API
implementation, 184
melody_pin0 and
melody_pin1 arrays, 188
non-blocking calls
code implementation, 187
materials list, 185
schematic diagram, 186
OUTPUT_FAST and INPUT_FAST, 184
void noTone, 185
void tone, 185
Tri-state buffers
diode, 215
logic states, 200
low-enabled gate operation, 201
NTE74HC125, 199
74HC125 and pin-out disposition, 200
Twitter application
access level, 300
BATT terminal, 292
coin battery, 293
create my access token, 299
create token and consumer keys, 301
creating account, 294
Linux shell script, 289
POST methods
bash code, 310
cat command, 310
curl command, 308
debug messages, 315
identify API, 301
input data session, 304
message header, 312
mPCIe module, 310
OAuth signing results, 305
OAuth string codes, 314
OAuth tool, 303, 311
popen() function, 315
post() function, 315
request settings, 304
should_post_today(), 315
signature_key, 311
soil_moisture_
with_twitter.ino, 316
system() function, 315
terminal shell, 305
tweet text messages, 311
twitter.sh, 308
twitter_sketch.ino, 312
sha1 algorithm, 289
signing application, 297
system and hardware clock, 290
temporary and
dirty workaround, 293–294
WiFi list, 290
USB-OTG adaptor, 582
uvcvideo module driver, 326

Video4Linux (V4L), 327
Video4Linux 2 (V4L2), 327
Virtual machine (VM), 101

Web page
  home automation, 470
  home.html, 472
  invokes socket, 474
  jQuery library version 2.1.0, 474
  socket connection, 471
  switch relays, 470
  temperature and PIR sensors, 470
txtsensor, 471

Web server creation
  cheerio installation, 469
  code writing
    final code, 481
    GET and POST methods, 478
    mywebserver.js, 474
    ports defining, 475
    running node.js, 475
    sockets creation, 476
    node.js updating, 465
    npm, 467
    socket.io installation, 469
    web page
      (see Web page)
WiFi mini-PCIe cards, 218
write() method, 148–149

Yocto build system
  class file, 43
  code structure, 38
  configuration file, 42
  Poky, 36
    (see also Intel Galileo images)
  recipe file, 40
YUYV encode, 330

ZMODEM protocol, 277