

Index

■ A

Advanced Audio Coding (AAC), 262
Application programming interface (API), 325
Audio and video rendering, 299–300
Audio_decode
 component, 221
 output device, 214
 PCM, 213–214
Audio decoder, 300–322, 324
Audio formats, 204–213
Audio port information
 audio_render, 95
 coding, 95–97
 fields, 93
 MP3 files, 94
 OMX_AUDIO_CODINGTYPE, 94
 OMX_AUDIO_PARAM_PCMMODETYPE, 97
 OMX_AUDIO_PARAM_PORTFORMATTYPE, 95
 OMX_AUDIO_PORTDEFINITIONTYPE, 93
 OMX_IndexParamAudioPcm, 97
 OMX_PARAM_PORTDEFINITIONTYPE, 93–94
 programs, 98–110
 structure, 93
 video_render, 111
Audio stream rendering, 263–266, 268–275
Avconv, 221–226
av_dump_format, 266
av_find_best_stream, 235
av_read_frame, 265
AV_SAMPLE_FMT_FLTP, 265
AV_SAMPLE_FMT_S16P, 235
av_samples_alloc, 265

■ B

Beethoven.mp3, 235, 244
Broadcom GPU
 components, 247
 threads, 247

Building programs

EGL initialization, 28–30
Makefile, 27
rendering context, 30–32

■ C

C99/C11 compilers, 10
Color codings, 157
Command-line parameter, 226
Compile command, 226
CreateSimpleTexture2D, 63, 247, 251

■ D

Debian-derived systems, 263
Debugging, 261
Demuxer, 275
Dispmanx, 14
 GPU accessing, 21–24
 layers, 375–376
 program, 21
doc/examples/decoding_encoding.c
 program, 221

■ E

EGL, 14
 drawing surface, 33–37
 specification, 27
EGLClientBuffer, 246
EglCreateImageKHR, 246
EGLImage buffer, 253
esLoadTGA, 62
esUtil functions
 esCreateWindow, 43
 esInitContext, 43
 esMainLoop, 43
 esRegisterDrawFunc, 43
 fields, 43

■ INDEX

esUtil functions (*cont.*)

- OS-dependent layers, 43
- struct, 43
- UserData, 46
- X Window System, 43

Event handler callback, 122

■ F

FFmpeg, 221-227, 229-239, 241-244, 263

FilledBuffer, 253

■ G

Garbage collection, 24

- Java, 37
- Khronos, 38

gcc Compile Flags, 11-12, 14

gcc version, 10

■ H

Hello_Triangle.c, 53

H.264 file decoding, 186-188, 190-193

■ I

ID3 extensions, 234-239, 241-244

IL Client library

- building programs, 129
- changing component state, 132-134
- client creation, 131
- debugging clients, 135-136
- public functions, 130-131
- waiting for events, 135

ilclient_set_fill_buffer_done_callback, 253, 255

il_enable_port_buffers, 207

il_ffmpeg_demux_render_audio_video.c, 301

■ J

JPEG image, 157-158

■ K

Khronos Group

- EGL, 8
- graphics and video, 7
- OpenGL ES, 7-8
- OpenMAX, 8
- OpenVG, 8

■ L

LibAV decoder, 226-227, 229-239,
241-244, 263-266, 300

libavcodec-extra, 235

Libraries

- Dispmanx, 15
- EGL, 15
- OpenGL ES, 16
- OpenMAX, 15
- OpenVG, 16

■ M, N

Mediainfo, 262, 266

Mp3 file

- decoding, 221-226
- rendering, 226-227, 229-234
- ID3 extensions, 234-239, 241-244

Multimedia files

- Big Buck Bunny, 262
- container files, 261
- decoding, 263-266
- demuxing, 263
- formats, 261

■ O

OMX_AllocateBuffer, 138, 168, 207

OMX audio decoder, 300

OMX.broadcom.video_decode, 275

omxplayer, 261, 263, 280, 300

OMX_TIME_CONFIG_SCALETYPETYPE, 280

OpenGL ES program, 7, 15, 246

animation

- esUtil, 71
- invoking, 72
- matrices, 71
- uniform parameters, 72-77

API, 42

building programs, 41-42

colored triangle

- passing multiple attributes,
vertex shader, 58
- squares drawing and other shapes, 59-62
- varyings concept, 57

Dispmanx, 41

drawing, 51-52, 376-380, 382-383, 385-386

esUtil functions, 43-44, 46

files, 41

opaque red triangle drawing, 53-54, 56

- OpenMAX video, 387–394, 396, 398
- program object, 50–51
- RPi, 41
- shaders
 - GPU, 47
 - loading, 49
 - minimal fragment, 48
 - minimal vertex, 48
- shading language, 48
- Silicon Graphics, 42
- surface, 375
- textures, 247–248, 250–251, 253–255, 257–259
 - attributes and shaders, 63–64
 - coding, image drawing, 67–71
 - drawing, 65–66
 - init function, 64
 - mipmaps, 63
 - object creation, 63
 - TGA files, 62
- vertices, 46
- OpenMAX, 153, 226–227, 229–239, 241–245, 252
 - API, 14–15
 - audio processing
 - audio formats, 204–213
 - building programs, 203
 - components, 204
 - broadcom, 80
 - buffers, 80
 - coding, 119–121
 - component states, 80, 113–114, 276
 - audio port information, 93, 95–97, 99–111
 - getting and setting parameters, 89
 - getting port information, 90–92
 - listcomponents, 84–86
 - Makefile, 85–86
 - OMX.broadcom.visualisation, 83
 - OMX_ComponentNameEnum(), 84
 - OMX_GetHandle(), 83, 87
 - OMX_GetRolesOfComponent, 84
 - OMX_init(), 84
 - port information, 88–89
 - port types, 92
 - Raspberry Pi, 86–87
 - specification, 83
 - hComponent, 122
 - idle to executing state, 122
 - image
 - OpenGL ES surface, 376–380, 382–383, 385–386
 - OpenVG, 398–407
 - layers, 79
 - loaded to idle
 - components, 115
 - nonworking program, 115–118
 - Makefile, 113
 - mechanisms, 80
 - multithreaded processing model, 80
 - OMX_StateLoaded to OMX_StateIdle, 118
 - ports, 80, 119
 - programming model, 80
 - specification, 5
 - state changes, 122, 124–128
 - state transition diagram, 114
 - transition to idle state, 122
 - waitFor, client thread, 128
 - wakeUp, event handler, 128
- OpenMAX buffers, Raspberry Pi
 - allocation, 138
 - building programs, 137
 - components, 137
 - EOS flag, 144
 - IL client library, 138, 140–142
 - image decoding, 146–148, 150–152
 - port settings, 145
 - sequence of actions, 143
 - video_decode, 138
 - writing and reading
 - OMX_EmptyThisBuffer, 142
 - OMX_FillBufferDone, 142
 - Video Core/Posix threads, 143
- OpenMAX video processing, 387–394, 396, 398
 - building programs, 181
 - H.264 File decoding, 186–188, 190–193
 - OpenVG, 408–418
 - Pango, 419–423, 425–431, 433
 - video components, 181
 - video formats, 182–186
- OpenVG, 8, 15
 - building programs, 325
 - Dispmanx and EGL, 326–330
 - drawing, 398–418
 - images
 - EGL context, 344
 - eglCreatePbufferFromClientBuffer, 345
 - image-user-to-surface transformation, 342
 - off-screen buffer image, 344
 - path, 345
 - pbuffer, 344
 - RGBA characteristics, 344
 - sources, 342
 - stride, 341
 - surface, 345
 - VGImage, memory data, 342

OpenVG (*cont.*)

- Mesa project, 326
- pink triangle, 331–336
- pipeline, 330
- specification
 - E-book readers, 326
 - games, 326
 - low-level graphics device interface, 326
 - portable mapping applications, 326
 - scalable user interfaces, 326
 - SVG and Adobe Flash viewers, 325
- standard shapes, 337–341

Output device, 214

■ **P, Q**

- Pango API, 419–423, 425–431, 433
- Pulse code modulated (PCM)
 - ADPCM data, 212, 214
 - analog signals, 213
 - avconv, 214
 - description, 213
 - encoding, 214
 - f32 and f64, 213
 - functions, 215–220
 - unencoded audio, 213
- pkg-config, 18–19
- Posix threads, 127
- pthreads, 247

■ **R**

- Raspberry Pi (RPi)
 - building programs, 153
 - communication, 153
 - CPU level, 3
 - ELinux.org, 4
 - GPU programming, 4
 - image components, 153
 - image formats, 154–157
 - JPEG Image, 157–158
 - Linux system, 3
 - Model B, 1
 - Pi Zero, 1
 - programming style, 5
 - rendering an image (without tunneling),
 - 168–169, 171, 173–174, 176–180
 - single-board computer, 1
- Rendering an image
 - without tunneling, 168–169, 171,
 - 173–174, 176–178, 180
 - with tunneling, 162, 164–168

■ **S**

- Sample Makefiles
 - Dispmanx, 16
 - EGL, 17
 - OpenMAX, 17
 - OpenVG, 18
- Screen capture, 25

■ **T, U**

- Text processing, OpenVG
 - building programs, 349
 - displaying text, 349
 - drawing text
 - Cairo, 350–354
 - complexities, 350
 - issues, 350
 - Pango, 355–360
 - FreeType
 - API reference, 361
 - bitmap, 367
 - font faces, 361–362
 - GPU, 361
 - Linux systems, 361
 - outline, 363, 365–366
 - packages, 349
 - paths and glyphs, 362–363
 - vgDrawpath, 366
 - glyph metrics, 371–372
 - vgCreateFont, 368
 - vgDrawGlyphs, 368–371
- Tunneling
 - broadcom components, 159, 160
 - function, 161
 - IL client, 158
 - ilclient_setup_tunnel, 160
 - image_decode, 160
 - proprietary communication, 158
 - pseudo-code, 159, 161–162
 - rendering an image, 162–168

■ **V**

- VCOS thread system, 128
- vgImageSubData, 342
- Video stream rendering
 - with full speed, 275–277
 - with scheduling
 - behind-the-scenes, 280
 - 64-bit timestamp, 284
 - clock component, 279, 282

clock setup, 280–281
 code, 278–280
 complete program, 284–287, 289–291, 293–299
 dts and pts, 283
 FFmpeg/LibAV, 281
 field, 283
 LibAV/FFmpeg, 283
 numerator (num) and
 denominator (den), 283
 OMX timestamps, 284
 OpenMAX components, 278

parameters, 280
 pts values, 283
 pts * timebase, 284
 substantial layers, 277
 tunnels, 281
 xScale, 280
 Vulkan graphics system, 42

■ **W, X, Y, Z**

WAV header, 300