

Index

- <- wiring operator, 31, **51**
- > wiring operator, 31, **51**
- = wiring operator, 31, **52**
- @ (declaring, using attributes), 142
- @C attribute, 143
- @atleastonce attribute, 143
- @atmostonce attribute, 143, 212
- @atomic_hwevent attribute, 143
- @combine attribute, 66, 143
- @exactlyonce attribute, 143
- @hwevent attribute, 143
- @integer attribute, 132, 143
- @number attribute, 133, 143
- @spontaneous attribute, 143
- #define, 41, 45
 - pitfalls, 46
- #include, usage of, 45
- __attribute__ (deprecated), 144
- 802.15.4, 5

- abstract data type, 133
 - implemented using generic modules, 133
 - implemented using reference parameters, 134
 - predefined in TinyOS, 247
- active message type, 89, 94, 98, 113, 242
- Active messages (AM)
 - address and TOS_NODE_ID, difference, **90**
- active messages (AM), 89, 113, 241
- ActiveMessageC component, 137
- Alarm interface, 231
- AM type, *see* active message type
- AMSend interface, 90
- AntiTheft (application), 79
- application source code, 9
- as
 - naming components, 53
 - naming interfaces, 23
- asynchronous (async)
 - code, **71**, 192
 - commands and events, **71**, 192
 - consistency issues, 134
 - use to minimize jitter, 228
- Atm128AdcSingle interface, 232
- atomic statement, **195**
 - execution time, 196
 - implemented by disabling interrupts, 195
 - limitations, 200
 - use in SoundLocalizer, 231
- attribute, **142**
- auto-wiring (for initialization), **60**, 139

- backing array (for packets), **114**, 115, 123, 124
- base station, **95**, 98
- bidirectional interface, **28**
- big-endian, 43
- binary reprogramming, 245
- BlockRead interface, 108
- BlockWrite interface, 106
- Boot interface, 81

- C libraries, using, 47
- C++ templates vs generic components, 130, 132
- callback, 11, 15, 167
- CC1000 (radio chip), **49**
- CC2420 (radio chip), **5**
- collection (network protocol), 95, 96, 98, 241
- combine function, **66**
 - associative and commutative, 67
 - warning (when missing), 67
- command, 11, **25**
 - unique, *see* unique (compile-time function)
 - uniqueCount, *see* uniqueCount (compile-time function)
- component, 6, 11, 14, **21**
 - implementation, 21, **29**
 - initialization, 59
 - layering, 60
 - libraries, 162
 - naming using as, 53
 - signature, **21**
 - singleton, **33**, 70
 - switching between similar, 94
- concurrency model, **71**, 192
- ConfigStorage interface, 103
- configuration, 12, 31, **50**
 - enum and typedef in body, 150
 - exporting interfaces, **52**
 - generic, 68, 130, **150**

- configuration (cont.)
 - use in Facade pattern, 183
 - use in Placeholder pattern, 180
- constants, 41
- Counter interface, 226
- CRC (cyclic redundancy check), 250
-
- data race, **193**
 - automatic detection, 196, 229, 231
 - avoiding, 196
- deadline-based timing, 81
- Deluge (binary reprogramming), 245
- design patterns, 166
 - Adapter, **189**
 - behavioral, 166, 186, 189
 - Decorator, **186**
 - Dispatcher, **166**
 - Facade, **183**
 - Keymap, **177**
 - Keyspace, **174**
 - namespace, 174, 177
 - Placeholder, **180**
 - Service Instance, **170**
 - structural, 170, 180, 183
- device driver, 210
 - access control, 211
 - and the Resource interface, 214
 - dedicated, 211
 - latency, 213
 - power management, 215
 - shared, 211
 - virtualized, 211
- dissemination (network protocol), 95, 97, 98, 241
- DisseminationUpdate interface, 99
- DisseminationValue interface, 97
- dynamic memory allocation, 37, 248
-
- energy management, *see* power management
- enum
 - in configuration, 150
 - use and abuse, 41
 - use in Global Keyspace pattern, 174
- event, 11, **25**
- execution model, **71**
- exporting (an interface), **52**
-
- fan-in, **64**
- fan-out, **64**
- FlashSampler (application), 105
-
- GeneralIO interface, 235
- generic component, **33**, 68, 129, 150
 - implemented by code copying, 130
 - type arguments, 132
 - use in Adapter pattern, 189
 - use in Decorator pattern, 187
 - use in Dispatcher pattern, 169
 - use in Service Instance pattern, 171
 - vs C++ template, 130, 132
- generic interface, **27**
- global declarations, 45
-
- hardware abstraction architecture (HAA), **206**, 221
 - storage, 208
 - timers, 209
- hardware adaptation layer (HAL), **206**
- hardware interface layer (HIL), **206**, 241
- hardware presentation layer (HIL), **207**
- header files, 45
- HIL, *see* hardware abstraction architecture
- HIL, *see* hardware adaptation layer (HAL)
-
- I²C, 235
- I2Cpacket interface, 236
- IEEE 802.15.4, 5
- interface, 11, 14, **24**, 156
 - bidirectional, **28**
 - default command, event handlers, 142
 - generic, **27**
 - naming using as, 23
 - parameterized, *see* parameterized interface
 - provider, 11, **22**
 - split-phase, **35**, 84
 - type, **56**
 - type parameter, **27**, 29
 - user, 11, **22**
- interrupt, 71, 192
- interrupt handler, 71, 192
 - and stack usage, 38
-
- keyspace, 152, 157, 160, **174**
-
- Leds interface, 80
- little-endian, 43
- low-power listening (radio), 216, 251
- LowPowerListening interface, 216, 251
-
- malloc, problems with, 37
- Matchbox filing system, 183
- McuParamOverride interface, 218
- McuParamState interface, 219
- McuParamSleep interface, 218
- memory
 - allocation, 37, 248
 - buffer swap in Receive, 39
 - conserving with enum, 41
 - ownership, **39**, 91, 93
 - sharing across components, 39, 41
- Message (Java class), 115
- message_t, **89**
- micaz mote, **5**

- mig (PC tool), 114
 - and AM types, 118
 - generated methods, 115
 - receiving packets, 117
 - sending packets, 116
 - module, 10, **30**
 - generic, **33**, 131, 150, 164
 - variable initializers, 33
 - mote, **4**
 - micaz, 5
 - power budget, 7
 - Telos, 5
 - MoteIF (Java class), 116, 121
 - and AM types, 118
 - receiving packets, 117
 - sending packets, 116
 - Mount interface, 103
 - multi-hop networking, 95
 - collection, *see* collection (network protocol)
 - dissemination, *see* dissemination (network protocol)
 - multiple wiring, **63**
 - order of calls, 63
 - naming convention
 - for components, 58
 - ncg (PC tool), 118, 176
 - and #define, 119
 - nesC
 - comparison with C and C++, 16
 - compilation model, 13
 - nesC reference manual, 7
 - nesdoc, 12, 61
 - net.tinyos.message
 - Message class, *see* Message (Java class)
 - MoteIF class, *see* MoteIF (Java class)
 - networking
 - multi-hop, 95
 - serial-port, *see* serial-port networking
 - single-hop, 89
 - norace, 229
 - nx_, nxle_ type prefixes, **43**
- packet
- reception, 93
 - sending, 90
 - size, **89**
 - specified with platform-independent types, 90, 112
 - structure, 89
 - structure (serial), 112, 113
 - unreliable delivery (radio), 89
 - unreliable delivery (serial), 112
- packet source (PC), 116, **119**
- parameterized interface, **137**, 145, 154, 156
 - and default handlers, 142
 - and dynamic dispatch, 140
 - and unique, 146
 - implementation in nesC, 139
 - in modules, 140
 - use in Dispatcher pattern, 167
 - use in Keymap pattern, 177
 - use in Keyspace pattern, 175
 - use in power lock implementation, 202
 - use in Service Instance pattern, 171
- PC tools, 112
 - Java, 112
 - mig, 114
 - MoteIF, *see* MoteIF (Java class)
 - ncg, 118, 176
 - other languages, 112
 - platform-independent types, 112
 - serial forwarder, **120**
- permanent storage, 101
 - block data, 106
 - configuration data, 103
 - log data, 108
 - reliability, 101, 103, 110
 - sampling to, 106
 - volume configuration file, 102
 - volumes, 102, 176
- platform-independent types, **43**
- platform-independent types
 - accessed using mig, 114
 - PC tools, 112
 - use in networking, 90, 97, 112
- portability, 206, 210, 221, 224
- posting a task, **72**
- power lock, **200**
 - arbiter, 202
 - configurator, 202, 204
 - library, 205
 - power manager, 202
 - split-phase access, 200
 - use in SoundLocalizer, 231
- power management, 251
 - for the microcontroller, 218
 - in dedicated drivers, 216
 - in device drivers, 215
 - in shared drivers, 217
 - in the radio, 216, 251
 - in virtualized drivers, 217
 - scheduler interaction, 218
 - using power locks, 203
- printf, 250
- race condition, **193**, *see* data race
- random number generation, 249
- Read interface, 84
- ReadStream interface, 87

- Receive interface, 93
- recursion
 - avoiding, 37
 - due to events signaled in commands, 76
 - use tasks to avoid, 77
- reliable transmission (serial), 120
- Resource interface, 201
- ResourceConfigure interface, 204
- ResourceDefaultOwnwr interface, 203
- RootControl interface, 100

- sampling sensors, 84
- sampling to permanent storage, 106
- Send interface, 96
- sending packets, 90
- sensor, 84
 - components, 85
 - stream sampling, 87
 - values, calibration, 86
- sensor networks, 3
 - structure (typical), 95
- sensor node, 4
- serial forwarder (PC tool), 120
- serial-port networking, 113, 241
- service
 - starting and stopping, 92, 97
- signature (of component), 21
- single stack, 35, 37
- single-hop networking, 89
- singleton (component), 33, 70
- SoundLocalizer (application), 221
- source code, for example applications, 9
- split-phase, 6, 34, 84
- SplitControl interface, 92
- stack usage, 37
- StdControl interface, 97
- synchronous (sync) code, 71, 192

- task, 6, 71, 72, 192
 - for avoiding recursion, 77
 - posting from interrupt handlers, 199
 - posting overhead, 74
 - split-phase operations, 75
 - timing, effects on latency and throughput, 74
- Telos mote, 5
- time synchronization, 222, 225
- Timer interface, 81
- TinyOS
 - 1.x, 134, 144, 211
 - API, 7, 241
 - compiling, 8
 - enhancement proposal (TEP), 241
 - installing, 8
 - overview, 5
 - stack, 35, 37
 - task scheduler, 72
 - TinyOS component
 - ActiveMessageC, 94, 243
 - AMReceiverC, 94, 242
 - AMSenderC, 94, 242
 - AMSnooperC, 242
 - AMSnoopingReceiverC, 242
 - BigQueueC, 248
 - BitVectorC, 247
 - BlockStorageC, 106, 246
 - CollectionC, 101
 - CollectionControlC, 243
 - CollectionReceiverC, 243
 - CollectionSenderC, 98, 243
 - ConfigStorageC, 102, 246
 - ConstantSensorC, 246
 - CrcC, 250
 - DemoSensorC, 246
 - DisseminationC, 101, 244
 - DisseminatorC, 98, 101, 244
 - FefsArbiterC, 205
 - LedsC, 83, 250
 - LocalTimeMilliC, 245
 - LogStorageC, 108, 246
 - MainC, 83, 241
 - McuSleepC, 218
 - NoLedsC, 250
 - PoolC, 69, 248
 - PrintfC, 250
 - QueueC, 247
 - RandomC, 249
 - RandomLfsrC, 249
 - RandomMlcrC, 249
 - RoundArbiterC, 205
 - sensors, 85, 245
 - SerialActiveMessageC, 94, 243
 - SerialAMReceiverC, 94, 243
 - SerialAMSenderC, 94, 243
 - SineSensorC, 246
 - StateC, 249
 - TimerMilliC, 83, 245
- TinyOS interface
 - Alarm, 231
 - AMSend, 90
 - BlockRead, 108
 - BlockWrite, 106
 - Boot, 81
 - ConfigStorage, 103
 - Counter, 226
 - DisseminationUpdate, 99
 - DisseminationValue, 97
 - GeneralIO, 235
 - I2CPacket, 236
 - Leds, 80
 - LowPowerListening, 216, 251
 - McuPowerOverride, 218
 - McuPowerState, 219

- McuSleep, 218
- Mount, 103
- Read, 84
- ReadStream, 87
- Receive, 93
- Resource, 201
- ResourceConfigure, 204
- ResourceDefaultOwnnewr, 203
- RootControl, 100
- Send, 96
- SplitControl, 92
- StdControl, 97
- Timer, 81
- TOS_NODE_ID, 59, 91
- TOSH_DATA_LENGTH, **89**
- tree collection, *see* collection
- type
 - big-endian, 43
 - little-endian, 43
 - of interfaces, **56**
- typedef
 - in configuration, 150
- unique (compile-time function), **146**, 151, 161
 - use in Local Keyspace pattern, 174
 - use in power lock implementation, 202
 - use in Service Instance pattern, 171
- unique (compile-time) function
 - use in shared device drivers, 214
- uniqueCount (compile-time function), **147**
 - use in Local Keyspace pattern, 176
 - use in power lock implementation, 202
 - use in Service Instance pattern, 171
- virtualized
 - device, 211
 - service, 147, 156
- volume configuration file (for storage), 102
- wiring, 12, 14, **31**
 - code generated for, 51
 - fan-in, **64**
 - fan-out, **64**
 - multiple, **63**
 - omitting interfaces in, 56
 - only a metaphor, 65
 - use in Facade pattern, 183
 - use in Keymap pattern, 177
 - use in Placeholder pattern, 180