

INDEX

Symbols

- #define
 - code, 211–213
 - conditional code, 214
 - parameterized, 210
 - preprocessor, 119, 120, 121, 208
- #elif (preprocessor), 216
- #endif (preprocessor), 214, 215
- #error (preprocessor), 218
- #ifdef (preprocessor), 214, 215
- #ifdef UNDEF (remove code), 219
- #ifndef (preprocessor), 215
- #include (preprocessor), 7, 12, 209, 217, 270
- #pragma (preprocessor), 217, 218
- #pragma diagnostic (GCC), 218
- #warning (preprocessor), 218
- % (modulus), 54
- %c (printf format), 238, 239
- %d (printf format), 54, 238, 239
- %f (printf format), 238
- %l (printf format), 238
- %ld (printf format), 58
- %lx (printf format), 62
- %o (printf format), 62, 63, 238
- %p (printf format), 95
- %u (printf format), 62, 63
- %x (printf format), 61, 62, 238, 239
- c (compiler option), 45
- E (compiler option), 16, 208
- g3 (compiler option), 45
- ggdb (compiler option), 10
- mcpu=cortex-m0 (compiler option), 45
- mfloat-abi=soft (compiler option), 45
- mthumb (compiler option), 45
- o (compiler option), 10, 45
- O0 (compiler option), 45
- static (compiler option), 14
- Wa (compiler option), 12
- Wall (compiler option), 10, 45
- Wextra (compiler option), 10
- Wl (compiler option), 14
- .align (assembler directive), 187, 189
- .bss
 - assembler directive, 187, 189
 - section, 187, 189, 191
- .comm (assembler directive), 187
- .config section, 195, 202
- .data
 - assembler directive, 187, 189
 - section, 187, 191, 192
- .global (assembler directive), 187
- .isr_vector section, 190, 191, 194
- .rodata section, 13, 188, 191
- .section (assembler directive), 188
- .space (assembler directive), 187, 189
- .string (assembly directive), 13
- .text section, 188, 191
- .text.main section, 188, 193, 194
- .weak (assembler directive), 164
- .word (assembler directive), 187, 189
- __attribute__((packed)), 125, 126, 135
- __attribute__((unused)), 152
- __attribute__((weak)), 279
- __attribute__((section(...))), 195
- __disable_irq, 181
- __enable_irq, 181
- __HAL_RCC_USART2_CLK_DISABLE, 155, 202
- __HAL_RCC_USART2_CLK_ENABLE, 149, 155, 201
- _ebss (linker symbol), 192
- _edata (linker symbol), 192
- _estack (linker symbol), 192
- _sbss (linker symbol), 192
- _sdata (linker symbol), 192
- + (addition operation), 54

- & (ampersand)
 - address of operator, 94
 - AND operator, 68, 69, 74
- * (asterisk)
 - dereference operator, 94
 - multiplication operation, 54
 - pointer declaration, 94
- \\ (backslash), 7
- \r (carriage return), 7, 146, 254
- /* ... */ (comment), 8
- // (comment), 8
- { } (curly brackets), 7
- (decrement operator), 66
- / (division operator), 54
- \ " (double quotes), 7
- == (equals operator), 78
 - = vs. == bug, 78, 79
- ^ (exclusive OR operator), 69
- > (greater than operator), 78
- >= (greater than or equal operator), 78
- ++ (increment operator), 65
 - preprocessor issues, 210
 - ++x vs. x++, 66
- << (left shift operator), 69, 70
- < (less than operator), 78
- <= (less than or equal operator), 78
- \n (line feed or newline), 7, 146, 254
- != (not equals operator), 78
- ~ (NOT operator), 68, 69
- \0 (NUL), 101
- += operator, 65
- | (OR operator), 67, 68
- @param (parameter), 143
- >> (right shift operator), 70
- (subtraction operation), 54
- \t (tab), 7

A

- absolute address, 194
- absolute object file, 192
- Ac6 STM32 MCU Project, 37
- add, integer, 54
- adding node, single linked list, 227, 228
- addition, fixed-point, 263–265
- address, 13
 - absolute, 194
 - pointers, 94
 - printing, 98
 - sanitizer, 234, 235
- address of operator (&), 94
- advanced linker usage, 195
- align (assembler directive), 187, 189
- aligned data, access, 125–127
- alignment, integer, 123, 124
- allocate heap (malloc), 224, 225
- allocation, local variable, 109, 110
- alternatives, floating-point, 262
- ambiguity, if/else, 79
- analog pin, 41
- analysis
 - assembly language, 180
 - interrupt code, 177, 179
- AND operator (&), 68, 69, 74
- anonymous struct, 130
- anti-pattern, 88, 89
 - assignment in while, 89
 - empty loop, 89
- Apple
 - end of line, 146, 254
 - USB serial setup, 158
- apt-get, 4
- ar command, 275–278
- argc parameter, 250
- argument
 - command line, 249, 250
 - promotion, 61
- argv parameter, 250
- arithmetic, pointer, 97
- ARM Cortex-M0, xviii, 57
- ARM GCC compiler, 45
- arm-none-eabi-objcopy, 46
- arm-none-eabi-size, 46
- array, 91, 92
 - declaration, 92
 - illegal, 92
 - index, 92, 100
 - initialization, 93
 - overflow, 98–100
 - to pointer assignment, 97
- arrays and pointers, relationship
 - between, 97
- as, 11
- ASCII, 239
- assembler, 11, 12

- assembler directive
 - .align, 187, 189
 - .bss, 187, 189
 - .comm, 187
 - .data, 187, 189
 - .global, 187
 - .section, 188
 - .space, 187, 189
 - .weak, 164
 - .word, 187, 189
 - assembly language, 10, 12
 - analysis, 180
 - assembly listing, 13
 - assert statement, 134, 135
 - assigning variables, 56
 - assignment, struct, 128
 - assignment in while (anti-pattern), 89
 - asynchronous interrupts, 171
 - attribute
 - packed, 125, 126, 135
 - section, 195
 - unused, 152
 - weak, 279
 - automatic conversion, 258
- B**
- backslash (\), 7
 - bare metal, xvii
 - Basic Settings dialog, 23
 - baud, 145
 - rate, 148
 - big machines, 221, 222
 - .bin (file extension), 47
 - binary
 - file, 243, 244–246
 - numbers, 59–60
 - bit
 - checking, 72, 73
 - clearing, 72
 - constant, 71
 - defining, 71
 - manipulation, example of, 72–74
 - meaning, 70
 - operations, 67
 - pattern, 60
 - printing, 74
 - setting, 71, 72
 - bitmapped I/O, 71
 - bitwise AND (&), 68, 69, 74
 - bitwise invert (~), 68, 69
 - bitwise exclusive OR (^), 69
 - bitwise OR (|), 67, 68
 - blinking LED, 32
 - bootloader, 204
 - breakpoint, 49
 - Breakpoints window, 28
 - break statement, 87
 - bss
 - assembler directive, 187, 189
 - ideal memory model, 185, 186
 - section, 187, 189, 191
 - size, 46
 - buffer
 - circular, 172, 173
 - log, 178, 179
 - buffered file I/O, 222, 237–246
 - vs. raw I/O, 251
 - buffering, 246
 - line, 246
 - serial, 172
 - bug, = vs. ==, 78, 79
 - Build Project, IDE, 24, 44
 - button, 83
 - get state of, 86
 - user, 34
 - byte, 57
 - order, 131, 132
- C**
- call by value, 128, 129
 - call stack trace, 183, 184
 - camel case, 55
 - carriage return (\r), 7, 146, 254
 - cc1, 11
 - CFLAG (make macro), 9
 - chapters, 114
 - char, 91, 101
 - character
 - printf, 238, 239
 - transmit (serial), 150
 - characters and strings, 100–101
 - char constant, 101
 - char* initialization, 101
 - checking, bit, 72, 73

- C ideal memory model, 185
- circular buffer, 172, 173
 - interrupt, 173
 - sending, 173
- clean
 - make target, 16
 - project, 44
- clearing, bit, 72
- clock, 34
 - serial, 145, 146
- close, 252
- closing files, 247
- C Managed Build, 54
- CMSIS directory, 47
- code
 - instrumenting, 177
 - interrupt analysis, 179
 - macro, preprocessor, 211–212
- combination shaving brush and fountain pen, 162
- comm (assembler directive), 187
- command line arguments, 249, 250
- comment, 8
- comment out code, 218, 219
- COMMON section, 191, 194
- comm section, 188
- compilation
 - conditional, 214, 215
 - cross, 21
 - of modules, 268
 - native, 21
 - of a program, 5
- compiler, 3, 10–16
 - ARM, 45
- compiler flags, 10
 - c, 45
 - E, 16, 208
 - g3, 45
 - ggdb, 10
 - mcpu=cortex-m0, 45
 - mfloat-abi=soft, 45
 - mthumb, 45
 - o, 10, 45
 - O0, 45
 - static, 14
 - Wa, 12
 - Wall, 10, 45
 - Wextra, 10
 - Wl, 14
- complex data types, 117
- conditional compilation, 214, 215
- CONFIG section, 195, 196, 202, 203
- configuration items, multiple, 202
- Console tab, 44
- Console window, 26, 28, 44
- const, 71, 91, 114, 118, 209
 - declaration, 92
 - demonstration, 102
- constant
 - bit, 71
 - global, 186
 - string, 186, 188
- const char* const, 101, 102
- const volatile, 165
- continue statement, 88, 89
- copy file (buffered), 244
- core dumped (floating-point exception), 261
- cpp, 11
- Cppcheck, 283
- CR1 (Control Register 1), 166–169, 173, 175, 176
- create mode, 253
- creative theft, 282
- creative writing, 281
- cross-compilation, 21
- cross file type checking, 269
- curly brackets ({}), 7
- custom type, 132, 133

D

- data
 - initialized, 185
 - type, complex, 117
 - uninitialized, 185, 186
- data
 - assembler directive, 187, 189
 - ideal memory model, 185, 186
 - section, 187, 191, 192
 - size, 46
- deallocate heap (free), 225
- deallocation, local variable, 109, 110
- Debug configuration, 23
- debug connector (JTAG), 34, 35

- Debug* directory, 47
- debugger, 10, 26, 47, 50, 28, 48
 - external, 194
 - internal, 195
 - interrupt, 178, 179
 - JTAG, xviii, 1, 34–35, 49
 - local variable, 110
 - perspective, 27, 28, 48
 - pointers, 96
 - stack frame, 111
 - ST-LINK, 36
 - struct, 122
 - variables, 56
- dec (size), 46
- decimal, 59
- decision statements, 77
- declaration
 - array, 92
 - const, 92
 - pointer, 94
 - variable, 55
- decrement operator (--), 66
- Default_Handler, 164
- define
 - code, preprocessor, 211–213
 - conditional code, preprocessor, 214
 - parameterized, preprocessor, 210
 - preprocessor, 208
- defining bits, 71
- deleting node, linked list, 230
- demonstration, const, 102
- dereference operator (*), 94
- designated initializers, 127
- deterministic library, 278
- development board, 34
- diagnostic (GCC #pragma), 218
- directives, preprocessor, 10
- directory
 - CMSIS*, 47
 - Debug*, 47
 - HAL_Driver*, 47
 - inc*, 47
 - startup*, 47
- disable interrupt, 181
- divide, integer, 54
- divide by zero, 246
- division, fixed-point, 263–265
- dnf, 4
- double inclusion protection, 270
- double, printf, 238
- double quotes (\"), 7
- do/while trick, preprocessor, 213
- Doxygen, 40, 108, 143, 283
- dynamic memory (heap), 221, 223–235
 - problems, 233

E

- E (compiler option), 16
- ebss (linker symbol), 192
- Eclipse, 4, 19–31
- edata (linker symbol), 192
- .elf* (file extension), 47
- elif (preprocessor), 216
- else statement, 79
- embedded computer, xvii
- embedded struct, 133, 134
- empty loop (anti-pattern), 89
- Empty Project, 54
- enable interrupt, 181
- endif (preprocessor), 214, 215
- end of line, 146, 254
- enum (enumerated type), 117, 118
 - illegal values, 118, 119
 - preprocessor trick, 119–121
 - type, 132, 133
- equals bug, 78, 79
- equals operator (==), 78
- error (preprocessor), 218
- Error_Handler, 152, 199
- ErrorHandler, 147
- errors, 6
 - preprocessor, 209–211
- estack (linker symbol), 192
- event logging, 177, 178, 179
- evil gets, 241
- examples, importing, 31
- exclusive OR operator (^), 69
- executable, 11
 - file, 184
- Executables window, 28
- exiting a loop (break), 87
- extern, 268–271, 274
- external debugger, 194

F

- F5 (Step Into), 50
- F6 (Step Over), 29, 30, 49, 50
- F8 (Resume), 30
- factorial, 112, 113
- fclose, 245, 246
- fgets, 228–230, 241, 243
- field, struct, 121, 122
- file
 - binary, 243, 244–246
 - buffered copy, 244
 - closing, 242, 246
 - executable, 184
 - I/O, buffered, 237–246
 - main.c*, 46
 - object, 184
 - opening, 242–243
 - output.map*, 47
 - program, 184
 - startup_stm32f030x8.s*, 47, 51, 163, 193
 - stm32f0xx_it.h*, 47
 - stm32f0xxit.c*, 46
 - syscalls.c*, 46
 - Systemstm32f0xx.c*, 46
- FILE* declaration, 242, 245
- firmware
 - library, 37
 - upgrade, 185, 204
- fixed-point, 262–265
 - addition, 263–265
 - division, 263–265
 - multiplication, 263–265
 - subtraction, 263–265
- fixing interrupt problem, 181
- FLASH, 202, 203
- flash (memory), 184
- FLASH_EraseInitTypeDef, 197, 200
- flash memory programmer, xviii, 34, 43, 48, 195
- FLASH section, 196
- floating pin, 85
- floating-point
 - automatic conversion, 258
 - alternatives, 262
 - divide exception, 246

- exception, 261
- guard digit, 259
- hardware, 262
- implementation, 262
- infinity, 260, 261
- money, 260
- NaN, 260, 261
- normalization, 260
- numbers, 222, 257–265
 - precision, 260
 - problems, 259
 - rounding error, 259
 - subnormal, 260, 262
- floating vs. integer divide, 258
- flushing, 246
- fopen, 242, 245
- for loop, 82, 93
- for(;;) (loop forever), 40, 42, 43, 84, 126
- fprintf, 240, 243
- fread, 244, 245
- free, 225
 - NULL pointer, 226
- function. *See* procedure
- function pointer, typedef, 136
- fwrite, 244, 245

G

- GCC, 3, 5, 10, 11
 - address sanitizer, 234–236
 - arm-none-eabi-gcc, 45
 - diagnostic (#pragma), 218
 - installing, 4
 - L (library search), 277
 - options, 44
 - warning, suggest parentheses, 79
- gcc -E, 208
- general purpose input/output. *See* GPIO
- gets, 241
- gddb, 10
- global (assembler directive), 187
- global
 - constant, 186
 - initialized, 186, 187
 - initialized to zero, 186, 187
 - uninitialized, 186, 187
 - variable, 105, 106

- GPIO, 70, 71, 83–86, 149, 150
 - internal wiring, 86
 - pin, 41, 42
 - register, 42
- GPIOA, 150, 155, 202
- GPIO_AF1_USART2, 150, 155
- GPIO_InitTypeDef, 42, 43, 83, 84, 155
- GPIO_MODE_AF_PP, 150, 155
- GPIO_MODE_INPUT, 84
- GPIO_MODE_OUTPUT_PP, 42, 43, 153
- GPIO_NOPULL, 42, 43, 150, 155
- GPIO_PIN_2, 150, 155, 202
- GPIO_PIN_3, 150, 155, 202
- GPIO_PIN_SET, 86
- GPIO_PinState, 83, 84
- GPIO_PULLDOWN, 85
- GPIO_PULLUP, 85, 153
- GPIO_SPEED_FREQ_HIGH, 42, 43, 153
- GPIO_SPEED_FREQ_LOW, 150, 155
- Gray code, 60
- greater than operator (>), 78
- greater than or equal operator (>=), 78
- guard digit, 259

H

- HAL (hardware abstraction layer), 33, 39, 84
 - library, 41, 42, 47
 - modules, 267
 - namespace, 273
- HAL_Delay, 42, 43, 147, 153, 154, 167, 176, 198
- HAL_Driver (directory), 47
- HAL_FLASH_ERROR_NONE, 200
- HAL_FLASH_ERROR_PROG, 200
- HAL_FLASH_ERROR_WRP, 200
- HAL_FLASH_Lock, 197, 200
- HAL_FLASH_Program, 197, 200
- HAL_FLASH_Unlock, 196, 200
- HAL_FLASHEx_Erase, 197, 200
- HAL_GPIO_DeInit, 155, 202
- HAL_GPIO_Init, 42, 43, 83, 84, 153, 199, 201
- HAL_GPIO_ReadPin, 84, 86
- HAL_GPIO_TogglePin, 42, 43, 49, 153, 198
- HAL_GPIO_WritePin, 84, 153, 198

- HAL_Init, 41, 43, 47, 83, 147, 154, 176, 200
- HAL_OK, 148, 199, 200
- HAL_StatusTypeDef, 200
- HAL_UART_Init, 147, 148, 154, 199
- HAL_UART_MspDeInit, 202
- HAL_UART_MspInit, 149, 149, 201
- HAL_UART_STATE_RESET, 149
- HAL_UNLOCKED, 149
- hardware
 - initialization, 41
 - specification, 71
 - to struct, 134
- heap, 110, 186, 221, 223
- hello.c*, 5
- hello, hello, hello, 108
- Hello World, 3, 5–16, 21, 24, 101
 - ANSI C Project, 22
 - one-character version, 142–143
 - serial, 141, 147
 - serial interrupt, 161, 174–175
- hex (size), 46
- hexadecimal, 59
 - printf, 238, 239
- hidden variable, 107–108

I

- I/O
 - bitmapped, 71
 - buffered, 222
 - buffered file, 237–246
 - interrupt, 161, 162
 - polling, 161, 162
 - raw, 222, 249–255
- IDE, 19–28
 - Basic Settings dialog, 23
 - Build Project, 24
 - Console window, 26
 - debugging, 26
 - generated makefile, 30
 - Problems tab, 21
 - project creation, 22
 - Project Explorer, 21
 - project screen, 20, 21
 - Run Configurations dialog, 25
 - Select Configurations dialog, 23
 - starting, 20
 - text editor, 21

- IDE (*continued*)
 - view, 21
 - workspace, 20
- ideal memory model, 185
 - bss, 185, 186
 - data, 185, 186
 - text, 185, 186
- ifdef (preprocessor), 214–215
- ifdef UNDEF (remove code), 219
- if/else ambiguity, 79
- ifndef (preprocessor), 215
- if statement, 77–79
- illegal array index, 92
- importing examples, 31
- inc* directory, 47
- include (preprocessor), 7, 12, 209, 217, 270
- include file, 217
- inclusion, double protection, 217, 270
- increment operator (++), 65–66
- index
 - array, 92
 - wrong array, 100
- Infinite_Loop, 164
- infinite recursion, 113
- INFINITY (floating point), 260–261
- initialization
 - array, 93
 - char*, 101
 - hardware, 41
 - string, 101
 - struct, 127
 - UART, 147
 - of variables, 56–57
- initialized
 - data, 185
 - global variable, 186, 187
 - local static variable, 189
 - module only, 186
 - static local, 187
 - to zero global, 186, 187
- initializers, designated, 127
- inline procedure, 211
- installing GCC, 4
- instrumenting code, 177
- int, 56, 114
 - long, 57
 - long long, 58
 - short, 57
- int16_t, 61
- int32_t, 61, 62
- int64_t, 61, 62
- int8_t, 61
- integer
 - add, 54
 - alignment, 123–124
 - divide, 54
 - vs. floating divide, 258
 - modulus, 54
 - multiply, 54
 - printf, 238, 239
 - size, 57–59
 - subtract, 54
 - unsigned, 62
- integrated development environment.
 - See* IDE
- internal debugger, 195
- interrupt
 - circular buffer, 173
 - code, analyzing, 177, 179
 - data analysis, 165
 - debugging, 178
 - disable, 181
 - enable, 181
 - handler, 279
 - hell, 171
 - I/O, 161, 162
 - mapping, 170
 - problem, 177
 - fixing, 181
 - routines, 163–164
 - serial, 174–175
 - shared variable, 180
- interrupt (nonstandard keyword), 170
- interrupt and status register. *See* ISR
- int main(), 7
- invert bit, or NOT operator (~), 68, 69
- ioctl, 255
- isalpha, 103
- isnormal, 262

ISR (interrupt and status register),
150–153, 163, 166, 168,
174–175, 198
isr_vector section, 190, 191, 194

J

JTAG
debug connector, 34, 35
debugger, xviii, 1, 34–35, 49
jumpers, 36

L

last in, first out (LIFO), 111
l as variable name, 55
ld (linker), 11
LD1 (LED), 37
LD2 (LED), 37, 199
leak, memory, 233
learning to write, 281
LED, 83
 blink, 32
 LD1, 37
 LD2, 37, 199
 power, 34
 user, 34
LED2, 41, 42
LED2_GPIO_CLK_ENABLE, 42, 43,
 83–84, 199
LED2_GPIO_PORT, 42, 43, 49, 83–84,
 153, 198–199
led2_Init, 147, 154, 167, 176, 200
LED2_PIN; , 42–43, 49, 83–84, 153,
 198–199
left shift operator (<<), 69–70
less than operator (<), 78
less than or equal operator (<=), 78
libc.a, 193
libm.a, 193
library, 11, 193, 273–278
 building, 275–278
 deterministic, 278
 make, 275–277
 nondeterministic, 278
 STM firmware, 37

LIFO (last in, first out), 111
line
 buffering, 246–247
 endings, 146
 markers, 208
line feed *See* newline
linked list, 224, 226–233
 adding node, 227, 228
 printing, 229, 230
linker, 11, 14–15, 45, 183–204
 advanced, 195
 directive
 MEMORY, 196
 SECTIONS, 196
 library, 193
 LinkerScript.ld, 45
 map, 14, 15, 184, 185, 193–194
 memory configuration, 194
 nonstandard section, 190
 section
 CONFIG, 196
 FLASH, 196
 RAM, 196
 symbol
 _ebss, 192
 _edata, 192
 _estack, 192
 _sbss, 192
 _sdata, 192
 _sidata, 192
 LinkerScript.ld, 45
linking
 process, 191
 symbols, 183
Linux, 4
 end of line, 146, 254
 USB serial setup, 158
list
 linked, 226–233
 node, 226
listing, assembly, 13
local include, 270
local static
 initialized, 189
 uninitialized, 189

- local variable, 105–107
 - allocation, 109, 110
 - deallocation, 109, 110
 - debugger, 110
 - initialized static, 187, 189
 - uninitialized, 186
 - uninitialized static, 187
- log buffer, 178, 179
- logging, event, 178
- LOGO, memory section, 203–204
- logo storage, 203
- long int, 57
 - printf, 238
- long long int, 58
- loop
 - anti-patterns, 88–90
 - control, 87
 - exiting (break), 87
 - for, 82
 - forever, 40, 82
 - restarting (continue), 88, 89
 - while, 79, 80
- looping statements, 80–83
- lousy programming, 79
- lower layer (interrupt code),
 - 164–165, 179

M

- macOS, 4
 - USB serial setup, 158
- macro processor, 12, 207
- macros
 - code, preprocessor, 211–213
 - parametrized, 210
 - make, 9, 271, 272
 - simple, 208
- magic number, 93
- main, 7
- main.c*, 46
- make, 9, 10, 15
 - library, 275–277
 - macros, 9, 271, 272
 - phony target, 16
- make clean, 16
- Makefile*, 9, 15–16, 31, 47
 - IDE-generated, 30
 - library, 275–277
 - modules, 268, 271

- malloc, 224–225
 - running out of memory, 225
- map, linker, 14, 15, 184, 185, 193–194
 - memory configuration, 194
- memory
 - CONFIG section, 203
 - dynamic, 223
 - flash, 184, 191, 192, 194, 202–203
 - heap, 223
 - layout struct, 122–124
 - leak, 233
 - LOGO section, 203
 - model, 185
 - RAM, 183–185, 191, 192, 194, 202, 203, 223
 - stack, 223
- MEMORY (linker directive), 196
- microcontroller, 32–51
- Microsoft filenames, 243
- MinGW (Windows), 4
- mistakes, 6
 - preprocessor, 209–211
- mkdir, 5
- modular programming, 267–279
- module-only symbol
 - initialized, 186
 - uninitialized, 186
- modules, 267–268
- Modules window, 28
- modulus, integer, 54
- Morse code, 90, 145
- multiple configuration items, 202–203
- multiplication, fixed-point, 263–265
- multiply, integer, 54

N

- names, variable, 55
- namespace, 272
- naming, struct, 129
- NaN (Not a Number), 246, 260–261
- native compilation, 21
- nested vectored interrupt, 168
- newline (`\n`), 7, 146, 254
- node (list), 226
- nondeterministic library, 278
- nonstandard section, linker, 190
- normalization, floating-point, 260
- not equals operator (`!=`), 78

NOT operator (~), 68, 69
NUCLEO-F030R8, xix, 32–51
 setup, 35
NUL character (\0), 101
NULL pointer dereference, 169
numbers, 53
 fixed-point, 263–265
 floating-point, 257–265
 magic, 93
 representation, 59
numerical analysis, 260
NVIC_EnableIRQ, 167, 168, 176

O

0 as variable name, 55
O_BINARY, 252–255
objcopy (arm-none-eabi-objcopy), 46
object file, 11, 184
 absolute, 192
 relocatable, 192, 193
O_CREAT, 252, 253
octal, 59
 printf, 238
one-character version of Hello World,
 142–143
open flags
 O_BINARY, 252–255
 O_CREAT, 252, 253
 O_RDONLY, 252, 253
open function, 252, 253
opening files, 242
open mode, file, 243
 rb, 245
 wb, 245
open source tools, 282–284
operating system, 222
operations, bit, 67
operator
 ++, 65
 +=", 65
 --, 66
 address of (&), 94
 AND (&), 68, 69, 74
 decrement (--), 66
 dereference (*), 94
 equals (==), 78
 exclusive OR (^), 69
 greater than (>), 78

 greater than or equal (>=), 78
 increment (++), 65
 left shift (<<), 69–70
 less than (<), 78
 less than or equal (<=), 78
 NOT (~), 68, 69
 not equals (!=), 78
 OR (|), 67, 68
 right shift (>>), 70
 shorthand, 65

optimizer, 3
O_RDONLY, 252, 253
OR operator (|), 67, 68
Outline window, 28
output.map, 47
overflow, 63–64
 array, 98–100
oversampling (serial), 149

P

PA2 pin, 149, 150
PA3 pin, 149, 150
packed attribute, 125, 126, 135
parameter, pointer, 128, 129
parameterized macros, 210
parentheses warning, 79
permanent memory (flash), 195
perspective, debug, 27, 28, 48
phony target, make, 16
PIC processor, 170
pin
 analog, 41
 floating, 85
 GPIO, 41, 42
 PA2, 149, 150
 PA3, 149, 150
 pulldown, 85
 pullup, 85
pointer, 91
 arithmetic, 97
 scaling, 98
 assign from array, 97
 debugger, 96
 declaration, 94
 function, 136
 typedef, 136
 parameters, 128, 129
 printf (%p), 95

- pointer (*continued*)
 - size
 - STM32, 96
 - x86, 96
 - struct, 128
 - and thing, 94
 - variable panel, 96
- polling I/O, 161–162
- power connector, 34
- power LED, 34
- pragma (preprocessor), 217, 218
- pragma diagnostic (GCC), 218
- precision, floating-point, 260
- predefined files, 240
- preprocessor, 10, 12, 207–219
 - code macro, 211–213
 - continuation character, 211–213
 - #define, 208
 - code, 211–213
 - parameterized, 210
 - do/while, 213
 - #elif, 216
 - #endif, 214, 215
 - enum trick, 119–121
 - #error, 218
 - errors, 209–211
 - #ifdef, 214, 215
 - #ifndef, 215
 - #include, 209, 217
 - ++ issues, 210
 - line markers, 208
 - #pragma, 217, 218
 - symbol, 216
 - command line, 216
 - predefined, 217
 - tricks, 218
 - #warning, 218
- printf, 1, 7, 11, 12, 14, 54, 93, 238, 239
 - %c, 238, 239
 - %d, 238, 239
 - %f, 238
 - %l, 238
 - %ld, 58
 - %lx, 62
 - %o, 62, 63, 238
 - %p, 95
 - %u, 62, 63
 - %x, 61, 62, 238, 239
- printing bits, 74
- problems
 - dynamic memory, 233
 - floating-point, 259
 - interrupt, 171, 177
 - fixing, 181
 - use after free, 234
- Problems tab, IDE, 21, 28
- procedure, 105, 106, 108, 109
 - inline, 211
- program file, 184
- programming
 - modular, 267–279
 - style, 114
- project creation, IDE, 22
- Project Explorer, IDE, 21
- project screen, IDE, 20, 21
- project type, 37
- promotion, argument, 61
- protection
 - double include, 217
 - mode, 253
- pulldown pin, 85
- pullup pin, 85
- putchar, 142–143
- puts, 14, 101, 108
- PuTTY, 156–157

R

- RAM, 183, 202, 203, 223
 - memory, 184–185, 191–192, 194
- RAM section, 196
- ranlib command, 276–277
- raw I/O, 222, 249–255
 - vs. buffered I/O, 251
 - close, 252
 - ioctl, 255
 - open, 252, 253
 - open flags
 - O_BINARY, 252–255
 - O_CREAT, 252, 253
 - O_RDONLY, 252, 253
 - read, 252, 253
 - write, 252, 253
- rb (open mode), 245
- read, 252, 253
- READ (10), SCSI, 134
- reading data, buffered I/O, 241

- recursion, 112, 113
 - infinite, 113
- Red Hat, 4
- register
 - GPIO, 42
 - I/O, 150, 151
 - stack (rsp), 110
- Registers tab (debug view), 110
- Registers window, 28
- relative address, 13
- Release configuration, 23, 31
- relocatable object file, 14, 192–193
- representation of numbers, 59
- reset, 34, 41, 43
- restarting loop (continue), 88, 89
- Resume (F8), 30
- return, 8, 109
- right shift operator (>>), 70
- rodata section, 13, 188, 191
- rounding error, floating-point, 259
- RS-232 standard (serial), 149
- rsp stack register, 110
- Run Configurations dialog, 25
- running out of stack, 113
- running the program, 43
- RX signal (serial), 144

S

- sane programming, 106
- sanitizer, address, 234–235
- sbss (linker symbol), 192
- scaling, pointer arithmetic, 98
- scanf, 241
- scope, variable, 106–107
- screen program, 158
- SCSI (small computer system interface), 133–134
 - READ (10), 134
- sdata (linker symbol), 192
- section
 - bss, 187, 189, 191
 - .comm, 188
 - COMMON, 191, 194
 - .config, 195, 202
 - CONFIG, 195, 196
 - data, 187, 191, 192
 - FLASH, 196
 - .isr_vector, 190, 191, 194
 - RAM, 196
 - .rodata, 13, 188, 191
 - .text, 188, 191
 - .text.main, 188, 193, 194
- section (assembler directive), 188
- section attribute, 195
- SECTIONS linker directive, 196
- sections, nonstandard, 190
- segment, text, 184
- serial “Hello World,” 147
- serial clock, 145
- serial communication, 146
- serial I/O, 2, 34, 35, 141
 - buffering, 172
 - history, 145
 - input, 144
 - interrupt, 174–175
 - ISR, 150–153, 163, 166, 168, 174, 175, 198
 - output, 143
 - speed, 148
 - TDR, 150, 151, 153
 - transmit, 150, 162, 163
 - to USB, 156
- setting bit value, 71, 72
- shared variable, interrupt, 180
- shorthand operators, 65
- short int, 57
- sidata (linker symbol), 192, 192
- signal
 - RX (serial), 144
 - TX (serial), 143
- simple macro, 208
- Single Step (F6), 29, 30, 49, 50
- singly linked list, 226–233
 - adding node, 227, 228
 - printing, 229, 230
- size.c*, 58
- size command (arm-none-eabi-size), 46
- sizeof, 58, 95, 224
- SOC (system on a chip), xviii
- source file, 11
- Source window, 28, 29
- space (assembler directive), 187, 189
- speed, serial, 148
- SQLite, 284
- sscanf, 241
- s_sin.o*, 193

- stack, 110–112, 186, 223
 - frame, 109–112
 - register (`rsp`), 110
 - running out of, 113
 - trace, 183, 184
- Stack Trace window, 28
- start bit, 145, 146
- startup directory, 47
- startup_stm32f030x8.S*, 47, 163, 164, 190, 192, 193
- statement
 - `break`, 87
 - `continue`, 88, 89
 - `decision`, 77
 - `else`, 79
 - `if`, 77–79
 - looping, 79
- static
 - compiler option, 14
 - initialized local variable, 189
 - local initialized, 187
 - local uninitialized, 187
- `stderr` predefined file, 240
- `stdin` predefined file, 240
- stdint.h*, 61, 135
- `stdout` predefined file, 240
- Step Into (F5), 50
- Step Over (F6), 29, 30, 49, 50
- ST-LINK, 36
- STM32F030x4 processor, xviii
 - stm32f0xx.h*, 216
 - stm32f0xx_hal.c*, 50
 - stm32f0xx_it.h*, 47
 - stm32f0xxit.c*, 46
- STM32 firmware library, 37
- STM32 NUCLEO-F030R8, 32–51
 - setup, 35
- STM32 pointer size, 96
- STM HAL modules, 267
- stop bit, 145, 146
- string, 91, 100–101
 - constant, 186, 188
 - initialization, 101
- string (assembly directive), 13
- struct, 117, 122–130, 133–135
 - anonymous, 130
 - assignment, 128
 - debugger, 122
 - embedded, 133–134
 - field, 121–122
 - from hardware, 134
 - initialization, 127
 - memory layout, 122–124
 - naming, 129
 - packed, 125
 - padding, 123, 124
 - pointer, 128
 - `typedef`, 137
- structure. *See* struct
- subnormal floating-point numbers, 260, 262
- subtract, integer, 54
- subtraction, fixed-point, 263–265
- summing two variables, 57
- symbol
 - definition
 - command line, 216
 - preprocessor, 216
 - predefined, 217
 - weak, 164, 278–279
- synchronous communications, 146
- syscalls.c*, 46l
- system on a chip (SOC), xviii
- Systemstm32f0xx.c*, 46
- System Workbench for STM32, 4, 20–31, 33, 37–40, 46, 50, 54, 112, 191, 216, 282, 285–288

T

- tab character (`\t`), 7
- Tasks window, 28
- TDR (transmit data register), 150, 151, 153, 162, 163, 170, 175, 198
- teletype, 145, 146
- text (ideal memory model), 185, 186, 188, 191
- text (size), 46
- text editor, IDE, 21

- text.main section, 188, 193, 194
- text section, 185, 188, 191
- text segment, 184
- theft, creative, 282
- things and pointers to things, 94
- toupper, 103
- translation, hardware to struct, 134
- transmit
 - serial, 150
 - serial I/O, 162, 163
- transmit data register. *See* TDR
- transmit empty bit. *See* TXE bit
- transmit shift register (TSR), 162, 163
- tricks, preprocessor, 218
- turning off bit, 72
- two's complement, 64, 65
- TX, serial I/O, 143, 162, 163
- TXE bit, 151, 152, 198
- type
 - checking across files, 269
 - custom, 132, 133
 - enum, 117, 118, 132, 133
 - int, 56
 - struct, 117, 121, 122, 132, 133
 - union, 117, 130, 132, 133
 - variable, 55
- typedef, 117, 135
 - function pointer, 136
 - and struct, 136
- typeof, 220

U

- UART, 41, 146, 150
 - initialization, 147
- UART_ADVFEATURE_NO_INIT, 148, 154, 199
- UART_FLAG_TXE, 151–153, 163, 198
- UART_HandleTypeDef, 166, 201
- UART_HWCONTROL_NONE, 148, 154, 199
- UART_MODE_TX_RX, 148, 154, 199
- UART_ONE_BIT_SAMPLE_DISABLE, 148, 154, 199
- UART_OVERSAMPLING_16, 148, 154, 199
- UART_PARITY_NONE, 148, 154, 199
- UART_STOPBITS_1, 148, 154, 199

- UART_WORDLENGTH_8B, 148, 154, 199
- Ubuntu, 4
- ugly code removal, 215
- uint8_t, 62, 94, 118, 121, 122
- uint16_t, 62
- uint32_t, 62, 121, 122
- uint64_t, 62, 94
- unaligned data, access, 125–127
- uninitialized
 - data, 185, 186
 - global, 186, 187
 - local, 186
 - local static variable, 189
 - module only, 186
 - static local, 187
 - variables, 56
- union, 117, 132, 133
 - field assignment, 131
 - type, 130
- unistd.h*, 251
- universal asynchronous receive-transmitter. *See* UART
- unsigned int, 114
- unsigned integers, 62
- unused attribute, 152
- upgrade, firmware, 185, 204
- upper layer (interrupt code), 164, 165, 179
- USART. *See* UART
- USART2, 149, 154, 199, 202
- USART2_IRQHandler, 163, 164, 166, 168
- USART2_IRQn, 167, 176
- USART_CR1_IDLEIE, 168
- USART_CR1_PEIE, 168
- USART_CR1_RXNEIE, 168
- USART_CR1_TCIE, 168
- USART_CR1_TXEIE, 166–169, 173–176
- USART_CR1_TXNEIE, 168
- USART_ISR_CMF, 169
- USART_ISR_CTSIF, 169
- USART_ISR_FE, 169
- USART_ISR_IDLE, 169
- USART_ISR_NE, 169
- USART_ISR_ORE, 169

- USART_ISR_PE, 169
- USART_ISR_RXNE, 169
- USART_ISR_TC, 169
- USART_ISR_TXE, 166, 168, 169, 169,
174, 175
- USB
 - connector, 37
 - to serial, xviii, 141, 156
 - serial setup
 - Linux, 158
 - MacOS, 158
 - Windows, 156–157
 - use after free, 234
 - USER_BUTTON_GPIO_CLK_ENABLE, 83
 - user LED (LED2), 34, 41, 42
 - user push button, 34

V

- Valgrind, 234, 235, 283
- variable
 - assignment, 56
 - complex, 51
 - debugger, 56
 - declaration, 55
 - global, 105, 106
 - hidden, 107
 - initialization, 56, 57
 - initialized local static, 189
 - local, 105–107
 - allocation, 109, 110
 - deallocation, 109, 110
 - debugger, 110
 - names, 55
 - panel, 56, 96, 111
 - scope, 106, 107
 - size, 57

- type, 55
- uninitialized, 56
- uninitialized local static, 189
- Variables window, 28
- void, 109
- volatile, 152, 164

W

- Wa (compiler option), 12
- Wall (compiler option), 10
- warning (preprocessor), 218
- wb (open mode), 245
- weak (assembler directive), 164
- weak attribute, 279
- weak symbol, 164, 278–280
- Wextra (compiler option), 10
- while
 - assignment in, 89
 - loop, 79, 80
- Windows
 - terminal emulator (PuTTY),
156–157
 - USB serial setup, 156–157
- Wl (compiler option), 14
- word (assembler directive), 187, 189
- workspace, IDE, 20
- write, 252, 253
- writing data past the end, 234
- wrong array index, 100

X

- x86 pointer size, 96
- xcode-select, 4

Z

- zero, divide by, 246