

*Посвящается
моей жене Жаклин и моим детям*

Содержание

От издательства	15
Об авторе	16
Глава 1 • Введение.....	17
Текстовые соглашения	18
STM32F103C8T6	18
FreeRTOS	20
libopenstm3	21
Никаких Ардуино!.....	21
Никаких IDE	22
Фреймворк разработчика	23
Для кого написана эта книга?	23
Что вам потребуется?	23
Программатор ST-Link V2.....	24
Макетная плата	25
Провода DuPont (перемычки).....	25
Развязывающие конденсаторы 0.1 мкФ.....	27
Адаптер последовательного порта USB-TTL.....	28
Источник питания	29
Мелочи.....	30
Итоги	31
Глава 2 • Настройка программного обеспечения	32
Используемые соглашения о каталогах.....	32
Операционное ПО.....	32
ПО для книги	33
Клонирование с помощью учетной записи github.....	33
Анонимная загрузка с github	33
libopenstm3.....	33
Ядро FreeRTOS	34
Кросс-компилятор ARM.....	34
Создание программного обеспечения.....	37
Инструмент ST-Link	37
Установки для пользователей Windows	39
Учетная запись и пароль Linux	39
Обновление WSL.....	40
Запуск Ubuntu	40
Обновление Ubuntu	40
Установка зависимостей.....	41
Установка Windows usbipd.....	41
Утилита STM32 ST-LINK.....	43
Итоги	45

Глава 3 • Включаем питание и мигаем светодиодом.....	46
Питание	46
Стабилизатор +3.3 В	47
USB-питание и вывод +5 В.....	48
Питание +3.3 В.....	48
Правило одного источника питания	49
Общий провод	50
Сброс (RESET)	50
Быстрая проверка	50
ST-Link V2.....	51
Утилита st-flash	53
Чтение STM32.....	53
Запись flash.....	54
Стирание flash	55
Утилита Windows STM32 ST-LINK.....	55
Функция Info.....	55
Функция чтения Read.....	56
Функция записи Write	57
Функция стирания Erase.....	58
Итоги.....	58
Дополнительные источники	58
Глава 4 • GPIO.....	59
Сборка miniblink.....	59
Прошивка miniblink	60
Исходный код miniblink.c	60
API-интерфейс GPIO.....	63
Конфигурация GPIO	64
Входные режимы портов	66
Выходные режимы портов	67
Двухтактный выход.....	69
Выход с открытым стоком.....	69
GPIO_MODE_OUTPUT_*_MHZ	70
Выстраиваем уток в ряд.....	70
Входы GPIO	71
Цифровой выход, двухтактный	71
Цифровой выход, открытый сток.....	71
Характеристики GPIO.....	72
Пороги входного напряжения	73
Пороги выходного напряжения.....	73
Программные задержки.....	74
Проблемы с программной задержкой.....	74
Итоги.....	75
Упражнения	75
Глава 5 • FreeRTOS.....	77
Возможности FreeRTOS	77

Многозадачность	77
Очереди сообщений	78
Семафоры и мьютексы.....	79
Таймеры	79
Группы событий	80
Программа blinky2	80
Сборка и тестирование blinky2	84
Выполнение	84
Файл FreeRTOSConfig.h.....	84
Соглашение об именах FreeRTOS.....	86
Макроопределения FreeRTOS	87
Итоги	87
Упражнения	88
Глава 6 • USART	89
Периферийное устройство USART/UART.....	89
Асинхронные данные.....	89
USB-адаптеры последовательного порта	90
Подключение	91
Проект uart	93
Проект	96
Проект uart2	99
USART API.....	103
Включаемые файлы.....	105
Тактовые сигналы	105
Конфигурация	105
DMA.....	105
Прерывания	105
Статус ввод/вывод	106
Последовательность настроек.....	106
FreeRTOS.....	106
Задачи.....	106
Очереди	107
Итоги	108
Упражнения	108
Глава 7 • Последовательный порт USB	109
Проблема с USB на плате Blue Pill	109
Введение в USB	111
Каналы и конечные точки	111
Периферийное устройство последовательного порта USB.....	112
Последовательное USB-устройство в Linux	112
Последовательное USB-устройство в macOS	113
Последовательное USB-устройство в Windows.....	113
USB-выводы GPIO	114
Исходный демокод.....	114
Функция cdcacm_set_config()	115

Функция <code>cdc_control_request()</code>	116
Функция <code>cdcacm_data_rx_cb()</code>	117
Задача USB task.....	118
USB-прием	119
USB-отправка.....	119
Демопрограмма последовательного порта USB	120
Итоги	122
Дополнительные источники	122
Упражнения	123
Глава 8 • Флеш-память с интерфейсом SPI	124
Представление W25QXX.....	124
Шина последовательного периферийного интерфейса SPI	124
Сигнал <code>chip select</code>	126
Соединения и напряжение.....	126
Схема подключения SPI-флеш-памяти	126
Управление выводом <code>NSS</code>	127
Конфигурация STM32 SPI.....	129
Тактовая частота SPI	131
Режимы синхронизации SPI	132
Порядок битов и длина слова	134
Ввод-вывод через SPI	135
Чтение регистра <code>SR1</code>	135
Ожидание готовности	136
Чтение ID производителя.....	136
Запись флеш-памяти.....	137
Стирание флеш-памяти.....	139
Чтение флеш-памяти	141
Демопрограмма.....	142
Запуск демоверсии	143
ID производителя.....	147
Режим <code>Power Down</code>	147
Итоги.....	147
Дополнительные источники	147
Упражнения	148
Глава 9 • Оверлеи	149
Проблема компоновки	149
Раздел <code>MEMORY</code>	150
<code>ENTRY</code> и <code>EXTERN</code>	152
Раздел <code>SECTIONS</code>	152
<code>PROVIDE</code>	155
Перемещение данных	155
Определение оверлея	156
Код оверлея	157
Заглушки оверлея	158
Диспетчер оверлеев.....	159
VMA и адреса загрузки.....	159

Использование в коде имен, созданных компоновщиком	161
Функция диспетчера оверлеев	162
Функция-заглушка.....	163
Демопрограмма.....	163
Извлечение оверлеев.....	164
Демопрограмма (продолжение)	168
Ловушка при изменении кода	170
Итоги	170
Дополнительные источники	171
Упражнения	171
Глава 10 • Часы реального времени (RTC)	172
Демонстрационные проекты	172
RTC с одним прерыванием	172
Конфигурация RTC.....	173
Прерывание и настройка.....	174
Процедура обработки прерываний.....	175
Уведомление о задаче	177
Мьютексы	179
Демопрограмма.....	179
Соединения UART1	182
Запуск демопрограммы.....	182
Прерывание <code>rtc_alarm_isr()</code>	184
EXTI-контроллер.....	185
Итоги	186
Упражнения	187
Глава 11 • Интерфейс I2C	188
Шина I2C.....	188
Ведущий и ведомый	188
Старт и стоп	189
Биты данных.....	189
I2C-адрес	190
I2C-транзакции	191
PCF8574: расширитель GPIO	192
Подключение через I2C.....	193
Линия $\overline{INT}$ микросхемы PCF8574	194
Конфигурация PCF8574.....	195
Подключение нагрузки к GPIO микросхемы PCF8574	196
Формирование фронтов	197
Демосхема	197
Программное обеспечение I2C	200
Проверка готовности I2C.....	201
Запуск I2C.....	202
Запись I2C	203
Чтение I2C	204
Перезапуск I2C	204
Демопрограмма.....	205

Демосессия.....	207
Итоги.....	209
Упражнения	209
Глава 12 • Дисплеи OLED.....	210
OLED-дисплей.....	210
Конфигурация	211
Подключение дисплея.....	212
Свойства дисплея.....	213
Схема демонстрационного примера.....	214
AFIO.....	214
Графика.....	216
Растровое изображение	217
Запись растра	219
Программное обеспечение изображения вольтметра.....	219
Главный модуль.....	221
Демопрограмма.....	223
Итоги.....	224
Упражнения	224
Глава 13 • OLED с использованием DMA.....	225
Проблемы	225
Схема	225
Операция прямого доступа к памяти	225
Выполнение прямого доступа к памяти	226
Сигналы запроса DMA.....	226
Демопрограмма.....	228
Инициализация DMA	230
Запуск прямого доступа к памяти.....	230
Задача управления OLED SPI/DMA	231
Процедура DMA ISR.....	234
Перезапуск передачи DMA	235
Выполнение демопримера	236
Дальнейшие проблемы	237
Итоги.....	237
Упражнения	238
Глава 14 • Аналого-цифровое преобразование	239
Ресурсы STM32F103C8T6.....	239
Демопрограмма.....	240
Аналоговые входы PA0 и PA1	240
Конфигурация периферийного устройства АЦП	240
Запуск демопрограммы	243
Чтение АЦП.....	243
Аналоговые напряжения	246
Итоги.....	248
Упражнения	248
Дополнительные источники	248

Глава 15 • Дерево тактовых сигналов	249
Основы	249
RC-генераторы	250
Кварцевые генераторы	250
Питание генераторов	250
Часы реального времени (RTCCLK)	252
Сторожевой таймер	252
Системный источник (SYSCLK).....	252
SYSCLK и USB	254
Шина ANB	254
rcc_clock_setup_in_hse_8mhz_out_72mhz().....	255
Периферийные устройства APB1	258
Периферийные устройства APB2	259
Таймеры	259
Функция rcc_set_mco().....	259
Демопример HSI.....	260
Демопример HSE.....	261
Демонстрация частоты ФАПЧ/2.....	262
Итоги	262
Дополнительные источники	263
Упражнения	263
Глава 16 • Режим PWM таймера 2	264
PWM-сигналы	264
Таймер 2	265
PWM-цикл	268
Вычисление предварительного делителя таймера.....	269
Цикл 30 Гц	269
Подключение сервопривода.....	270
Запуск демопрограммы	271
PWM на PB3.....	271
Другие таймеры.....	272
Больше PWM-каналов	273
Итоги	273
Дополнительные источники	274
Упражнения	274
Глава 17 • PWM-вход с таймером 4	275
Сервосигнал	275
Напряжение сигнала.....	275
Демопроект	276
Конфигурация GPIO	276
Конфигурация таймера 4.....	276
Цикл task1	278
Процедура обработки прерывания ISR	278
Запуск демопрограммы	279
Вывод результатов сессии	281
Входы таймера.....	281

Итоги	283
Упражнения	283
Глава 18 • Шина CAN	284
Шина CAN	284
Дифференциальные сигналы	286
Доминантный/Рецессивный	287
Шинный арбитраж	288
Синхронизация	289
Формат сообщения	289
Ограничение STM32	290
Демопрограмма	290
Сборка программы	291
UART-интерфейс	291
Прошивка микроконтроллеров	292
Шина для демонстрации	292
Запуск демопрограммы	293
CAN-сообщения	296
Синхронизация	296
Итоги	296
Дополнительные источники	297
Глава 19 • Программное обеспечение CAN-шины	298
Инициализация	298
Функция <code>can_init()</code>	300
Приемные фильтры CAN	301
Прерывания приема CAN	302
Прием в приложении	305
Отправка CAN-сообщений	307
Итоги	308
Упражнения	308
Глава 20 • Прерывания UART	309
Основы	309
Подпрограммы обработки прерываний (ISR)	309
Демопрограмма	310
Разработка программы	310
Инициализация тактовых сигналов	312
Инициализация GPIO	312
Инициализация UART	312
Процедура обработки прерываний	313
Функция <code>uart_getc()</code>	314
Задача <code>main_task()</code>	315
Функция <code>uart_putc()</code>	315
Задача <code>tx_task()</code>	316
Запуск демопрограммы	317
Конфигурация	317
Запуск демопрограммы	318

Потеря данных	319
Итоги	320
Дополнительные источники	320
Упражнения	320
Глава 21 • Новые проекты	321
Создание проекта.....	321
Makefile	322
Дополнительные Makefile	324
Заголовки зависимостей.....	324
Параметры компиляции.....	324
Прошивка 128к.....	325
FreeRTOS.....	325
Модуль rtos/opencm3.c	325
Модуль rtos/heap_4.c.....	326
Необходимые модули.....	327
FreeRTOSConfig.h.....	327
Пользовательские библиотеки.....	329
Ошибки новичков	329
Итоги	330
Дополнительные источники	330
Упражнения	330
Глава 22 • Поиск неисправностей.....	331
Отладчик GNU (GDB).....	331
GDB-сервер	331
Удаленный GDB.....	332
Текстовый пользовательский интерфейс GDB.....	335
Проблемы с выводами GPIO периферии.....	336
Альтернативная функция не работает	337
Сбой периферии.....	338
Сбой прерывания во FreeRTOS	338
Переполнение стека	338
Оценка размера стека	339
Когда отладчик не помогает.....	340
Push/Pull (двухтактный выход) или открытый сток.....	340
Дефекты периферии	341
Ресурсы.....	341
Библиотека libopencm3	342
Приоритеты задач FreeRTOS	343
Планирование в libopencm3.....	344
Итоги	345
Приложение А • Ответы на упражнения	346
Приложение Б • Выводы GPIO STM32F103C8T6	356
Предметный указатель	359

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Об авторе

Уоррен Гей (Warren Gay) с детства увлекался электроникой и после школы часто таскал домой выброшенные телевизоры. В старшей школе он научился программировать IBM 1130, а затем продолжил карьеру в области разработки программного обеспечения в Политехническом институте Райерсона в Торонто. До выхода на пенсию он работал более 40 лет профессионально, в основном в области C/C++, Unix и Linux. Однако его любовь к электронике не угасала с момента создания самодельной системы Intel 8008 в 1970-х годах и до современных проектов на основе Raspberry Pi. Уоррен также имеет лицензию опытного радиолюбителя и в 1991 году получил возможность работать с космической станцией «Мир» (U2MIR) с использованием пакетной радиосвязи. Он является автором многих книг, в том числе «Sams Teach Yourself Linux in 24 Hours», «Linux Socket Programming by Example» и «Advanced Unix Programming».

Глава 1 • Введение

Сегодня существует значительный интерес к платформе ARM Cortex, поскольку устройства на основе технологий ARM можно встретить повсюду. Модули, содержащие ARM-основу, варьируются от небольших встроенных систем с микроконтроллерами до мобильных телефонов и крупных серверов под управлением Linux. Вскоре технологии ARM в больших количествах также будут присутствовать в центрах обработки данных. Все это веские причины ознакомиться с платформой ARM.

Даже несмотря на перспективы появления стандарта RISC-V, существуют причины, по которым платформа ARM будет продолжать оставаться важной. RISC-V – новая технология, многие расширения которой все еще находятся в разработке. Следовательно, производители будут пока стараться придерживаться курса ARM просто потому, что эти технологии проверены и известны затраты.

С учетом того что технологии варьируются от микроконтроллеров до полноценных серверов, естественно, возникает вопрос: «Зачем изучать программирование встроенных устройств? Почему бы не сосредоточиться на системах конечных пользователей под управлением Linux, таких как Raspberry Pi?»

Простой ответ заключается в том, что встроенные системы хорошо работают в сценариях, неудобных для более крупных систем. Они часто используются для взаимодействия с физическим миром, например для обмена данными между окружающей средой и настольной системой. В обычной клавиатуре используется специальный микроконтроллер (МК), который сканирует состояние клавиш и сообщает о событиях нажатия в настольную систему. Это не только уменьшает количество необходимых проводов, но и освобождает главный процессор от необходимости тратить свои высокопроизводительные вычисления на простую задачу отслеживания событий нажатия клавиш.

Другие приложения включают в себя встроенные системы в заводских цехах для систем контроля температуры, безопасности и обнаружения пожара. Нет смысла использовать полноценную настольную систему для такого типа приложений. Автономные встроенные системы экономят деньги и готовы к работе мгновенно. Наконец, небольшой размер МК делает его альтернативным выбором для летающих дронов, где вес является решающим фактором.

Разработка встроенных систем традиционно требовала специалистов двух дисциплин:

- инженера по аппаратному обеспечению;
- разработчика программного обеспечения.

Часто одному человеку поручают разработать конечный продукт. Инженеры по аппаратному обеспечению специализируются на разработке электронных схем, но в конечном итоге для продукта требуется программное обеспечение. Это может быть проблемой, поскольку специалистам по программированию обычно не хватает знаний в области электроники, а инже-

нерам часто не хватает опыта работы с программным обеспечением. Из-за сокращения бюджета и сроков поставки инженер-электронщик часто становится одновременно и инженером-программистом.

Нет ничего плохого в том, чтобы один человек выполнял оба аспекта проектирования, если имеются необходимые навыки. Независимо от того, являетесь ли вы инженером-электронщиком, разработчиком программного обеспечения, любителем или производителем, нет ничего лучше настоящей реальной практики, которая поможет вам начать работу. Вот о чем эта книга.

Текстовые соглашения

В этой книге используется ряд текстовых соглашений.

Код в тексте: указывает служебные слова, имена переменных в тексте, имена функций, названия программных библиотек. Пример: «Введите `exit`, чтобы выйти из сеанса».

Блок кода или команд задается следующим образом:

```
$ cd build/Release
$ sudo make install
```

Имена файлов, имена папок, расширения файлов, пути файловой системы выделяются курсивом. Пример: «В результате у вас должен остаться загруженный файл с именем *master.zip*». Также выделяются курсивом названия разделов на страницах сайтов, например: «Вы можете найти ссылку на вкладке *Source Code/Downloads*».

Названия в меню или диалоговых окнах выделены жирным шрифтом с русским переводом в скобках. Например: «Нажмите кнопку **Start** (Пуск)».

Названия клавиш на клавиатуре и их сочетаний также выделяются жирным шрифтом и заключаются в угловые скобки: **<Ctrl+X>**.



Так обозначаются советы.



Так обозначаются примечания общего характера.



Так обозначаются предупреждения и предостережения.

STM32F103C8T6

Для этой книги выбрано устройство STMicroelectronics STM32F103C8T6. Такое обозначение компонента кажется очень сложным, поэтому давайте разберем его:

- STM32 (платформа STMicroelectronics);
- F1 (семейство устройств);
- 03 (подраздел семейства устройств);
- C8T6 (исполнение, указывающее на объем SRAM, флеш-памяти и т. д.).

Как следует из названия платформы, эти устройства основаны на 32-битных вычислениях, т. е. значительно более мощные, чем 8-битные устройства.

F103 (F1+03) – одна из ветвей платформы STM32. Название определяет возможности центрального процессора (ЦП) и периферийных устройств.

Наконец, суффикс C8T6 дополнительно определяет возможности устройства, такие как объем памяти и тактовая частота.

Устройство STM32F103C8T6 было выбрано для этой книги по следующим причинам:

- низкая стоимость (на eBay можно приобрести всего за 2.4 долл. США),
- доступность (eBay, Amazon, AliExpress и т. д.),
- расширенные возможности,
- малогабаритный форм-фактор.

STM32F103C8T6 в течение ближайшего времени, вероятно, останется самым дешевым пособием для изучения платформы ARM Cortex-M3 как для студентов, так и для любителей. Устройство легкодоступно и достаточно функционально. Наконец, форм-фактор небольшой печатной платы позволяет припаивать разъемы по краям и подключать устройство к макетной плате – наиболее удобный способ проведения широкого спектра экспериментов.

Микроконтроллер на синей печатной плате (рис. 1.1) ласково называют Blue Pill («синяя таблетка»), по мотивам фильма «Матрица». Есть несколько старых печатных плат красного цвета, которые назывались Red Pill («красная таблетка»). Есть и другие, черные, известные как Black Pill («черная таблетка»). В этой книге я буду предполагать, что у вас есть модель синего цвета, как на рисунке. Помимо некоторых особенностей USB, между ней и другими моделями не должно быть особых различий.



Рис. 1.1. Печатная плата STM32F103C8T6 с припаянными контактными разъемами (Blue Pill)

Низкая стоимость имеет еще одно преимущество – она позволяет вам иметь несколько устройств, например для проектов, связанных с передачей

данных по CAN. В этой книге рассматривается связь CAN с использованием трех устройств, соединенных общей шиной. Низкая стоимость означает, что вы останетесь в рамках студенческого бюджета.

Поддержка периферии STM32F103 просто удивительная, если учесть его цену. Периферийные устройства в комплекте включают:

- четыре 16-битных порта входов/выходов общего назначения GPIO (General Purpose Input/Output), большинство из них устойчивы к напряжению 5 В,
- три порта USART (Universal Synchronous/Asynchronous Receiver/Transmitter – универсальный синхронный/асинхронный приемник/передатчик),
- два интерфейса I2C,
- два интерфейса SPI,
- два аналого-цифрового преобразователя (АЦП),
- два контроллера прямого доступа к памяти (DMA),
- четыре таймера,
- сторожевой таймер,
- USB-контроллер,
- CAN-контроллер,
- генератор контрольной суммы CRC (Cyclic Redundancy Check, контрольный избыточный код),
- 20 Кбайт статической оперативной памяти,
- флеш-память 64 (или 128) Кбайт,
- процессор ARM Cortex M3, максимальная тактовая частота 72 МГц.

Однако есть некоторые ограничения. Например, контроллеры USB и CAN не могут работать одновременно. Другие периферийные устройства могут конфликтовать за использование контактов ввода-вывода. Большинство конфликтов контактов разрешаются с помощью конфигурации AFIO (Alternate Function Input Output, альтернативная функция ввода-вывода), позволяющей использовать разные контакты для функций периферийного устройства.

В конфигурации периферийных устройств можно индивидуально включать различные источники тактовых импульсов для настройки энергопотребления. Расширенные возможности этого микроконтроллера делают его удобным для изучения. То, что вы узнаете о семействе STM32F103, можно будет использовать в более продвинутых версиях, таких как STM32F407.

Официально указан объем флеш-памяти 64 Кбайта, но вы можете обнаружить, что устройство поддерживает 128 Кбайт. Эта особенность описана в главе 2 и позволяет размещать на устройстве приложения большого размера.

FreeRTOS

В отличие от популярного семейства чипов AVR (ныне принадлежащего Microchip¹), семейство STM32F103 имеет достаточный объем статической

¹ В 2016 году официально было объявлено о приобретении создателя семейства AVR фирмы Atmel компанией Microchip. – *Здесь и далее в сносках примечания переводчика.*

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru