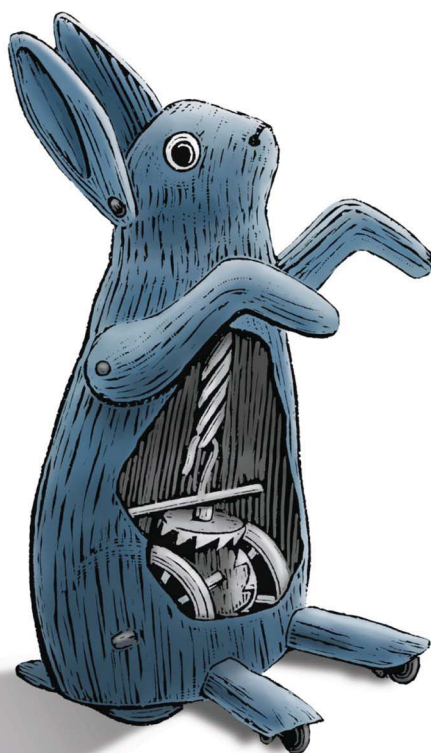


O'REILLY®

3-е издание

# Arduino

## БОЛЬШАЯ книга рецептов



Майкл Марголис  
Брайан Джепсон  
Николас Роберт Уэлдин

УДК 004.89  
ББК 32.973.26-018  
М25

**Марголис, М.**

М25 Arduino. Большая книга рецептов. — 3-е изд.: Пер. с англ. / М. Марголис, Б. Джепсон, Н. Р. Уэлдин. — СПб.: БХВ-Петербург, 2021. — 896 с.: ил.

ISBN 978-5-9775-6687-2

Этот справочник будет идеальным пособием для любого, кто хочет создавать проекты на базе популярной платформы Arduino. Он содержит подробное описание решений свыше 200 практических задач по созданию различных устройств и приспособлений, включая проекты для Интернета вещей, мониторинга окружающей среды, системы для определения местонахождения и положения в пространстве, а также устройств, которые реагируют на касание, звук, тепло и освещение. Все примеры третьего издания обновлены для версии 1.8 среды Arduino IDE с учетом современных концепций программирования. Каждое решение включает в себя программный код с подробными комментариями, его анализ и обсуждение возможных проблем. Книга будет полезна как начинающим, так и опытным разработчикам.

*Для радиолобителей*

УДК 004.89  
ББК 32.973.26-018

**Группа подготовки издания:**

Руководитель проекта	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Людмила Гауль</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Оформление обложки	<i>Карины Соловьевой</i>

© 2021 BHV

Authorized Russian translation of the English edition of *Arduino Cookbook*, 3E ISBN 978149190352

© 2020 Michael Margolis, Nicholas Robert Weldin, and Brian Jepson

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Авторизованный перевод с английского языка на русский издания *Arduino Cookbook*, 3E ISBN 978149190352

© 2020 Michael Margolis, Nicholas Robert Weldin, Brian Jepson.

Перевод опубликован и продается с разрешения компании-правообладателя O'Reilly Media, Inc.

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

ISBN 978-1-491-90352-0 (англ.)  
ISBN 978-5-9775-6687-2 (рус.)

© Michael Margolis, Nicholas Robert Weldin, Brian Jepson, 2020  
© Перевод на русский язык, оформление.  
ООО "БХВ-Петербург", ООО "БХВ", 2021

---

# Оглавление

<b>Предисловие</b> .....	<b>1</b>
Для кого предназначена эта книга? .....	1
Организация книги.....	2
Что не вошло в эту книгу? .....	5
Запись кода программ.....	6
Примечания к третьему изданию книги.....	7
Стили типографского оформления .....	8
Использование примеров кода.....	9
Онлайновая программа обучения O'Reilly .....	9
Контакты.....	10
Благодарности от Майкла Марголиса за второе издание .....	10
Благодарности от Брайана Джемсона за третье издание .....	12
<b>Глава 1. Начало работы</b> .....	<b>13</b>
1.0. Введение .....	13
Скетчи и программное обеспечение Arduino .....	14
Аппаратная часть платформы Arduino.....	14
1.1. Установка интегрированной среды разработки Arduino IDE.....	18
ЗАДАЧА .....	18
РЕШЕНИЕ .....	18
Установка среды Arduino IDE на Windows.....	18
Установка среды Arduino IDE на macOS .....	19
Установка среды Arduino IDE на Linux.....	20
Возможные проблемы установки .....	22
Дополнительная информация .....	22
1.2. Подготовка платы Arduino к работе .....	23
ЗАДАЧА .....	23
РЕШЕНИЕ .....	23
Обсуждение возможных проблем .....	24
Дополнительная информация .....	25
1.3. Создание скетча в среде разработки Arduino IDE.....	25
ЗАДАЧА .....	25
РЕШЕНИЕ .....	25
Обсуждение работы скетча и возможных проблем .....	28
Дополнительная информация .....	29
1.4. Загрузка и исполнение скетча Blink .....	29
ЗАДАЧА .....	29
РЕШЕНИЕ .....	29
Обсуждение работы скетча и возможных проблем .....	30
Дополнительная информация .....	31

1.5. Создание и сохранение скетча .....	31
ЗАДАЧА .....	31
РЕШЕНИЕ .....	31
Обсуждение работы скетча и возможных проблем .....	32
1.6. Простой первый проект Arduino .....	35
ЗАДАЧА .....	35
РЕШЕНИЕ .....	35
Обсуждение работы скетча и возможных проблем .....	37
Дополнительная информация .....	41
1.7. Работа с платами Arduino, неподдерживаемыми по умолчанию средой Arduino IDE .....	41
ЗАДАЧА .....	41
РЕШЕНИЕ .....	41
Обсуждение работы решения и возможных проблем .....	44
Дополнительная информация .....	45
1.8. Работа с 32-разрядными (или совместимыми с ними) платами Arduino .....	45
ЗАДАЧА .....	45
РЕШЕНИЕ .....	46
Обсуждение работы примера и возможных проблем .....	48
Дополнительная информация .....	49
<b>Глава 2. Программирование на языке Arduino .....</b>	<b>51</b>
2.0. Введение .....	51
2.1. Типичный скетч Arduino .....	52
ЗАДАЧА .....	52
РЕШЕНИЕ .....	52
Обсуждение работы решения и возможных проблем .....	53
Дополнительная информация .....	54
2.2. Простые типы данных (переменные) .....	54
ЗАДАЧА .....	54
РЕШЕНИЕ .....	54
Обсуждение работы решения и возможных проблем .....	56
Дополнительная информация .....	58
2.3. Работа с числами с плавающей запятой (точкой) .....	58
ЗАДАЧА .....	58
РЕШЕНИЕ .....	58
Обсуждение работы решения и возможных проблем .....	59
Дополнительная информация .....	61
2.4. Работа с группами значений .....	61
ЗАДАЧА .....	61
РЕШЕНИЕ .....	61
Обсуждение работы решения и возможных проблем .....	63
Дополнительная информация .....	65
2.5. Работа со строками в языке Arduino .....	66
ЗАДАЧА .....	66
РЕШЕНИЕ .....	66
Обсуждение работы решения и возможных проблем .....	67
Выбор между строками типа String и строками из массивов символов .....	69
Дополнительная информация .....	71
2.6. Использование массива символов .....	71
ЗАДАЧА .....	71
РЕШЕНИЕ .....	72
Обсуждение работы решения и возможных проблем .....	73
Дополнительная информация .....	73

2.7. Разбиение на группы текста, разделенного запятыми .....	74
ЗАДАЧА .....	74
РЕШЕНИЕ .....	74
Обсуждение работы решения и возможных проблем .....	75
Дополнительная информация .....	77
2.8. Преобразование числа в строку .....	77
ЗАДАЧА .....	77
РЕШЕНИЕ .....	77
Обсуждение работы решения и возможных проблем .....	77
2.9. Преобразование текста в число .....	79
ЗАДАЧА .....	79
РЕШЕНИЕ .....	80
Обсуждение работы решения и возможных проблем .....	81
Дополнительная информация .....	82
2.10. Организация кода в функциональные блоки .....	83
ЗАДАЧА .....	83
РЕШЕНИЕ .....	83
Обсуждение работы решения и возможных проблем .....	86
Дополнительная информация .....	87
2.11. Возвращение функцией нескольких значений .....	87
ЗАДАЧА .....	87
РЕШЕНИЕ .....	87
Обсуждение работы решения и возможных проблем .....	90
2.12. Условные операции .....	91
ЗАДАЧА .....	91
РЕШЕНИЕ .....	91
Обсуждение работы решения и возможных проблем .....	92
Дополнительная информация .....	93
2.13. Циклическое исполнение последовательности операторов .....	93
ЗАДАЧА .....	93
РЕШЕНИЕ .....	93
Обсуждение работы решения и возможных проблем .....	94
Дополнительная информация .....	95
2.14. Повторение исполнения кода с использованием счетчика .....	95
ЗАДАЧА .....	95
РЕШЕНИЕ .....	95
Обсуждение работы решения и возможных проблем .....	96
Дополнительная информация .....	98
2.15. Преждевременное завершение цикла .....	99
ЗАДАЧА .....	99
РЕШЕНИЕ .....	99
Обсуждение работы решения и возможных проблем .....	100
Дополнительная информация .....	100
2.16. Реагирование на множественные состояния одной переменной .....	100
ЗАДАЧА .....	100
РЕШЕНИЕ .....	100
Обсуждение работы решения и возможных проблем .....	102
Дополнительная информация .....	102
2.17. Сравнение символьных и числовых значений .....	103
ЗАДАЧА .....	103
РЕШЕНИЕ .....	103
Обсуждение работы решения и возможных проблем .....	105
Дополнительная информация .....	106

2.18. Сравнение строк .....	106
ЗАДАЧА .....	106
РЕШЕНИЕ .....	106
Обсуждение работы решения и возможных проблем .....	106
Дополнительная информация .....	107
2.19. Логические сравнения .....	107
ЗАДАЧА .....	107
РЕШЕНИЕ .....	108
Обсуждение работы решения и возможных проблем .....	108
2.20. Операции с битами .....	108
ЗАДАЧА .....	108
РЕШЕНИЕ .....	109
Обсуждение работы решения и возможных проблем .....	110
Дополнительная информация .....	111
2.21. Комбинирование операции и присваивания .....	112
ЗАДАЧА .....	112
РЕШЕНИЕ .....	112
Обсуждение работы решения и возможных проблем .....	112
Дополнительная информация .....	112
<b>Глава 3. Математические операции .....</b>	<b>113</b>
3.0. Введение .....	113
3.1. Сложение, вычитание, умножение и деление .....	113
ЗАДАЧА .....	113
РЕШЕНИЕ .....	113
Обсуждение работы решения и возможных проблем .....	113
Дополнительная информация .....	115
3.2. Увеличение и уменьшение значений переменных .....	115
ЗАДАЧА .....	115
РЕШЕНИЕ .....	115
Обсуждение работы решения и возможных проблем .....	115
Дополнительная информация .....	116
3.3. Нахождение остатка от деления двух значений .....	116
ЗАДАЧА .....	116
РЕШЕНИЕ .....	116
Обсуждение работы решения и возможных проблем .....	117
Дополнительная информация .....	118
3.4. Определение абсолютного значения числа .....	118
ЗАДАЧА .....	118
РЕШЕНИЕ .....	118
Обсуждение работы решения и возможных проблем .....	119
Дополнительная информация .....	119
3.5. Ограничение диапазона значений числа .....	119
ЗАДАЧА .....	119
РЕШЕНИЕ .....	119
Обсуждение работы решения и возможных проблем .....	119
Дополнительная информация .....	120
3.6. Определение меньшего или большего числа из двух или более чисел .....	120
ЗАДАЧА .....	120
РЕШЕНИЕ .....	120
Обсуждение работы решения и возможных проблем .....	120
Дополнительная информация .....	121

3.7. Возведение числа в степень .....	121
ЗАДАЧА .....	121
РЕШЕНИЕ .....	122
Обсуждение работы решения и возможных проблем .....	122
3.8. Извлечение квадратного корня .....	122
ЗАДАЧА .....	122
РЕШЕНИЕ .....	122
Обсуждение работы решения и возможных проблем .....	122
3.9. Округление с повышением и понижением чисел с плавающей запятой.....	123
ЗАДАЧА .....	123
РЕШЕНИЕ .....	123
Обсуждение работы решения и возможных проблем .....	123
3.10. Работа с тригонометрическими функциями .....	124
ЗАДАЧА .....	124
РЕШЕНИЕ .....	124
Обсуждение решения и возможных проблем .....	124
Дополнительная информация .....	125
3.11. Генерация произвольных чисел .....	125
ЗАДАЧА .....	125
РЕШЕНИЕ .....	125
Обсуждение работы решения и возможных проблем .....	126
Дополнительная информация .....	129
3.12. Установка и считывание значений битов .....	129
ЗАДАЧА .....	129
РЕШЕНИЕ .....	129
Обсуждение работы решения и возможных проблем .....	132
Дополнительная информация .....	132
3.13. Смещение битов .....	133
ЗАДАЧА .....	133
РЕШЕНИЕ .....	133
Обсуждение работы решения и возможных проблем .....	133
Дополнительная информация .....	134
3.14. Извлечение старшего или младшего байта из значений типа <i>int</i> и <i>long</i> .....	134
ЗАДАЧА .....	134
РЕШЕНИЕ .....	134
Обсуждение работы решения и возможных проблем .....	135
Дополнительная информация .....	136
3.15. Создание целого числа типа <i>int</i> или <i>long</i> из отдельных байтов .....	136
ЗАДАЧА .....	136
РЕШЕНИЕ .....	137
Обсуждение работы решения и возможных проблем .....	137
Дополнительная информация .....	139
<b>Глава 4. Последовательная связь.....</b>	<b>141</b>
4.0. Введение .....	141
Аппаратные средства последовательной связи .....	143
Особенности поведения аппаратных последовательных портов.....	145
Эмуляция аппаратного последовательного порта посредством цифровых контактов ввода/вывода .....	147
Протоколы обмена сообщениями .....	147
Замечания по реализации последовательной связи в скетчах Arduino .....	149

4.1.	Передача данных скетчем Arduino на компьютер.....	150
	ЗАДАЧА .....	150
	РЕШЕНИЕ .....	150
	Обсуждение работы решения и возможных проблем .....	151
	Дополнительная информация .....	154
4.2.	Отправка форматированного текста и числовых данных с платы Arduino.....	155
	ЗАДАЧА .....	155
	РЕШЕНИЕ .....	155
	Обсуждение работы решения и возможных проблем .....	156
	Дополнительная информация .....	159
4.3.	Прием последовательно передаваемых данных платой Arduino .....	159
	ЗАДАЧА .....	159
	РЕШЕНИЕ .....	159
	Обсуждение работы решения и возможных проблем .....	161
	Дополнительная информация .....	165
4.4.	Передача с платы Arduino нескольких текстовых строк в одном сообщении .....	165
	ЗАДАЧА .....	165
	РЕШЕНИЕ .....	165
	Обсуждение работы решения и возможных проблем .....	167
	Дополнительная информация .....	173
4.5.	Прием платой Arduino нескольких текстовых полей в одном сообщении .....	173
	ЗАДАЧА .....	173
	РЕШЕНИЕ .....	174
	Обсуждение работы решения и возможных проблем .....	175
	Дополнительная информация .....	176
4.6.	Отправка двоичных данных с платы Arduino .....	177
	ЗАДАЧА .....	177
	РЕШЕНИЕ .....	177
	Обсуждение работы решения и возможных проблем .....	178
	Размер переменной .....	181
	Порядок байтов .....	182
	Синхронизация .....	182
	Управление потоком данных .....	183
	Дополнительная информация .....	183
4.7.	Прием на компьютере двоичных данных с платы Arduino .....	183
	ЗАДАЧА .....	183
	РЕШЕНИЕ .....	183
	Обсуждение работы решения и возможных проблем .....	185
	Дополнительная информация .....	186
4.8.	Передача двоичных значений скетчем Processing на плату Arduino .....	186
	ЗАДАЧА .....	186
	РЕШЕНИЕ .....	186
	Обсуждение работы решения и возможных проблем .....	189
4.9.	Передача выходных значений нескольких контактов Arduino .....	190
	ЗАДАЧА .....	190
	РЕШЕНИЕ .....	190
	Обсуждение работы решения и возможных проблем .....	191
	Дополнительная информация .....	194
4.10.	Сохранение отправляемых Arduino данных в файл на компьютере.....	195
	ЗАДАЧА .....	195
	РЕШЕНИЕ .....	195
	Обсуждение работы решения и возможных проблем.....	197
	Дополнительная информация .....	199



4.11. Отправка данных с платы Arduino на несколько устройств с последовательным интерфейсом .....	199
ЗАДАЧА .....	199
РЕШЕНИЕ .....	199
Обсуждение работы решения и возможных проблем.....	203
Дополнительная информация .....	204
4.12. Прием данных платой Arduino по последовательному интерфейсу от нескольких периферийных устройств .....	204
ЗАДАЧА .....	204
РЕШЕНИЕ .....	204
Обсуждение работы решения и возможных проблем.....	208
Прием данных по нескольким программным портам.....	208
Дополнительная информация .....	211
4.13. Использование Arduino с Raspberry Pi .....	211
ЗАДАЧА .....	211
РЕШЕНИЕ .....	211
Обсуждение работы решения и возможных проблем.....	214
<b>Глава 5. Простой ввод цифровых и аналоговых данных .....</b>	<b>217</b>
5.0. Введение .....	217
5.1. Работа с кнопочными переключателями.....	222
ЗАДАЧА .....	222
РЕШЕНИЕ .....	222
Обсуждение работы решения и возможных проблем.....	224
Дополнительная информация .....	226
5.2. Подключение кнопки без внешних подтягивающих резисторов.....	227
ЗАДАЧА .....	227
РЕШЕНИЕ .....	227
Обсуждение работы решения и возможных проблем.....	229
5.3. Надежное определение нажатия кнопки (устранение дребезга контактов).....	229
ЗАДАЧА .....	229
РЕШЕНИЕ .....	229
Обсуждение работы решения и возможных проблем.....	231
Дополнительная информация .....	233
5.4. Определение длительности нажатия кнопки .....	233
ЗАДАЧА .....	233
РЕШЕНИЕ .....	233
Обсуждение работы решения и возможных проблем.....	236
5.5. Считывание сигналов с цифровой клавиатуры .....	239
ЗАДАЧА .....	239
РЕШЕНИЕ .....	239
Обсуждение работы решения и возможных проблем.....	242
Дополнительная информация .....	243
5.6. Считывание аналоговых сигналов .....	243
ЗАДАЧА .....	243
РЕШЕНИЕ .....	243
Обсуждение работы решения и возможных проблем.....	245
Дополнительная информация .....	245
5.7. Масштабирование значения к другому диапазону.....	246
ЗАДАЧА .....	246
РЕШЕНИЕ .....	246
Обсуждение работы решения и возможных проблем.....	247
Дополнительная информация .....	249

5.8. Считывание свыше шести входных аналоговых сигналов .....	249
ЗАДАЧА .....	249
РЕШЕНИЕ .....	249
Обсуждение работы решения и возможных проблем .....	251
Дополнительная информация .....	252
5.9. Измерение напряжений до 5 В .....	253
ЗАДАЧА .....	253
РЕШЕНИЕ .....	253
Обсуждение работы решения и возможных проблем .....	256
5.10. Реагирование на изменения напряжения .....	257
ЗАДАЧА .....	257
РЕШЕНИЕ .....	257
Обсуждение работы решения и возможных проблем.....	258
5.11. Измерение напряжений выше 5 В с помощью делителя напряжения.....	259
ЗАДАЧА .....	259
РЕШЕНИЕ .....	259
Обсуждение работы решения и возможных проблем.....	260
<b>Глава 6. Считывание данных с датчиков.....</b>	<b>263</b>
6.0. Введение .....	263
Дополнительная информация .....	265
6.1. Плата Arduino с несколькими встроенными датчиками .....	266
ЗАДАЧА .....	266
РЕШЕНИЕ .....	266
Обсуждение работы решения и возможных проблем.....	270
Дополнительная информация .....	270
6.2. Определение движений .....	270
ЗАДАЧА .....	270
РЕШЕНИЕ .....	270
Обсуждение работы решения и возможных проблем.....	272
Дополнительная информация .....	273
6.3. Определение уровня освещенности .....	274
ЗАДАЧА .....	274
РЕШЕНИЕ .....	274
Обсуждение работы решения и возможных проблем.....	275
Дополнительная информация .....	277
6.4. Определение движения живых существ .....	277
ЗАДАЧА .....	277
РЕШЕНИЕ .....	277
Обсуждение работы решения и возможных проблем.....	278
6.5. Измерение расстояния .....	279
ЗАДАЧА .....	279
РЕШЕНИЕ .....	279
Обсуждение работы решения и возможных проблем.....	281
Дополнительная информация .....	285
6.6. Точное измерение расстояния.....	286
ЗАДАЧА .....	286
РЕШЕНИЕ .....	286
Обсуждение работы решения и возможных проблем.....	288
Дополнительная информация .....	288
6.7. Определение вибраций .....	288
ЗАДАЧА .....	288
РЕШЕНИЕ .....	289
Обсуждение работы решения и возможных проблем.....	290

6.8. Обнаружение звука .....	290
ЗАДАЧА .....	290
РЕШЕНИЕ .....	290
Обсуждение работы решения и возможных проблем .....	292
6.9. Измерение температуры .....	296
ЗАДАЧА .....	296
РЕШЕНИЕ .....	296
Обсуждение работы решения и возможных проблем .....	298
Дополнительная информация .....	300
6.10. Чтение меток RFID (NFC) .....	301
ЗАДАЧА .....	301
РЕШЕНИЕ .....	301
Обсуждение работы решения и возможных проблем.....	302
6.11. Отслеживание вращательного движения .....	305
ЗАДАЧА .....	305
РЕШЕНИЕ .....	305
Обсуждение работы решения и возможных проблем.....	306
6.12. Отслеживание вращения вала кодера с использованием прерываний .....	308
ЗАДАЧА .....	308
РЕШЕНИЕ .....	308
Обсуждение работы решения и возможных проблем.....	309
Дополнительная информация .....	310
6.13. Работа с мышью .....	311
ЗАДАЧА .....	311
РЕШЕНИЕ .....	311
Обсуждение работы решения и возможных проблем.....	314
Дополнительная информация .....	315
6.14. Определение местонахождения с помощью системы GPS .....	316
ЗАДАЧА .....	316
РЕШЕНИЕ .....	316
Обсуждение работы решения и возможных проблем.....	318
Дополнительная информация .....	322
6.15. Определение вращения с помощью гироскопа .....	322
ЗАДАЧА .....	322
РЕШЕНИЕ .....	322
Обсуждение работы решения и возможных проблем.....	325
Дополнительная информация .....	325
6.16. Определение ориентации .....	326
ЗАДАЧА .....	326
РЕШЕНИЕ .....	326
Обсуждение работы решения и возможных проблем.....	328
6.17. Определение ускорения.....	329
ЗАДАЧА .....	329
РЕШЕНИЕ .....	329
Обсуждение работы решения и возможных проблем.....	331
Дополнительная информация .....	332
<b>Глава 7. Управление светодиодами устройства вывода данных .....</b>	<b>333</b>
7.0. Введение .....	333
Цифровой вывод .....	333
Аналоговый вывод.....	334

Управление световыми устройствами .....	335
Технические характеристики светодиодов .....	335
Мультиплексирование .....	337
Максимальный ток контакта .....	337
7.1. Подключение и использование светодиодов .....	338
ЗАДАЧА .....	338
РЕШЕНИЕ .....	338
Обсуждение работы решения и возможных проблем .....	340
Дополнительная информация .....	341
7.2. Управление яркостью светодиода .....	341
ЗАДАЧА .....	341
РЕШЕНИЕ .....	341
Обсуждение работы решения и возможных проблем .....	343
Дополнительная информация .....	343
7.3. Работа с мощными светодиодами .....	343
ЗАДАЧА .....	343
РЕШЕНИЕ .....	343
Обсуждение работы решения и возможных проблем .....	344
Как получить больше 40 мА на микроконтроллере Atmega? .....	345
Дополнительная информация .....	346
7.4. Управление цветом многоцветного светодиода .....	346
ЗАДАЧА .....	346
РЕШЕНИЕ .....	346
Обсуждение работы решения и возможных проблем .....	349
Дополнительная информация .....	350
7.5. Управление несколькими цветными светодиодами одним контактом платы Arduino .....	350
ЗАДАЧА .....	350
РЕШЕНИЕ .....	350
Обсуждение работы решения и возможных проблем .....	352
Дополнительная информация .....	353
7.6. Управление последовательностью светодиодов для создания линейного индикатора .....	354
ЗАДАЧА .....	354
РЕШЕНИЕ .....	354
Обсуждение работы решения и возможных проблем .....	356
Дополнительная информация .....	359
7.7. Управление последовательностью светодиодов для создания эффекта бегущего огня .....	359
ЗАДАЧА .....	359
РЕШЕНИЕ .....	360
Обсуждение работы решения и возможных проблем .....	361
7.8. Управление светодиодной матрицей с использованием мультиплексирования .....	361
ЗАДАЧА .....	361
РЕШЕНИЕ .....	361
Обсуждение работы решения и возможных проблем .....	364
7.9. Вывод изображения на светодиодную матрицу .....	366
ЗАДАЧА .....	366
РЕШЕНИЕ .....	366
Обсуждение работы решения и возможных проблем .....	368
Дополнительная информация .....	369
7.10. Управление светодиодной матрицей с использованием чарлиплексирования .....	370
ЗАДАЧА .....	370
РЕШЕНИЕ .....	370
Обсуждение работы решения и возможных проблем .....	372
Дополнительная информация .....	377

7.11. Управление 7-сегментным светодиодным дисплеем .....	377
ЗАДАЧА .....	377
РЕШЕНИЕ .....	377
Обсуждение работы решения и возможных проблем.....	380
7.12. Управление многоразрядным 7-сегментным светодиодным дисплеем с использованием мультиплексирования.....	380
ЗАДАЧА .....	380
РЕШЕНИЕ .....	380
Обсуждение работы решения и возможных проблем.....	383
7.13. Управление многоразрядным 7-сегментным светодиодным дисплеем с использованием минимального количества контактов .....	383
ЗАДАЧА .....	383
РЕШЕНИЕ .....	383
Обсуждение работы решения и возможных проблем.....	385
7.14. Управление светодиодной матрицей с использованием драйвера дисплея MAX72xx .....	386
ЗАДАЧА .....	386
РЕШЕНИЕ .....	386
Обсуждение работы решения и возможных проблем.....	388
Дополнительная информация .....	389
7.15. Увеличение количества выходных аналоговых контактов платы с использованием микросхемы расширения ШИМ-сигнала.....	389
ЗАДАЧА .....	389
РЕШЕНИЕ .....	389
Обсуждение работы решения и возможных проблем.....	391
Дополнительная информация .....	392
7.16. Использование в качестве дисплея аналогового измерительного прибора .....	392
ЗАДАЧА .....	392
РЕШЕНИЕ .....	392
Обсуждение работы решения и возможных проблем.....	393
Дополнительная информация .....	394
<b>Глава 8. Управление электродвигателями .....</b>	<b>395</b>
8.0. Введение .....	395
Сервомашинки .....	395
Соленоиды и реле .....	397
Щеточные и бесщеточные электродвигатели.....	397
Шаговые двигатели.....	398
8.1. Управление угловой позицией объекта с помощью сервомашинки.....	398
ЗАДАЧА .....	398
РЕШЕНИЕ .....	399
Обсуждение работы решения и возможных проблем.....	400
8.2. Управление сервомашинкой с помощью потенциометра или другого датчика.....	401
ЗАДАЧА .....	401
РЕШЕНИЕ .....	401
Обсуждение работы решения и возможных проблем.....	403
8.3. Управление скоростью вращения сервомашинки непрерывного вращения.....	403
ЗАДАЧА .....	403
РЕШЕНИЕ .....	403
Обсуждение работы решения и возможных проблем.....	406
8.4. Управление сервомашинками с компьютера.....	406
ЗАДАЧА .....	406
РЕШЕНИЕ .....	406

Обсуждение работы решения и возможных проблем .....	408
Дополнительная информация .....	408
8.5. Управление бесщеточным электродвигателем с помощью любительского контроллера ...	408
ЗАДАЧА .....	408
РЕШЕНИЕ .....	408
Обсуждение работы решения и возможных проблем .....	409
8.6. Управление соленоидами и реле.....	410
ЗАДАЧА .....	410
РЕШЕНИЕ .....	410
Обсуждение работы решения и возможных проблем .....	411
8.7. Вибрация объекта.....	412
ЗАДАЧА .....	412
РЕШЕНИЕ .....	412
Обсуждение работы решения и возможных проблем .....	413
8.8. Управление щеточным электродвигателем с использованием транзистора.....	415
ЗАДАЧА .....	415
РЕШЕНИЕ .....	415
Обсуждение работы решения и возможных проблем .....	416
8.9. Управление направлением вращения щеточного электродвигателя с помощью Н-моста .....	416
ЗАДАЧА .....	416
РЕШЕНИЕ .....	417
Обсуждение работы решения и возможных проблем .....	418
8.10. Управление направлением и скоростью вращения щеточного электродвигателя с помощью Н-моста .....	420
ЗАДАЧА .....	420
РЕШЕНИЕ .....	420
Обсуждение работы решения и возможных проблем .....	422
8.11. Управление направлением и скоростью вращения щеточного электродвигателя с помощью датчиков.....	423
ЗАДАЧА .....	423
РЕШЕНИЕ .....	423
Обсуждение работы решения и возможных проблем.....	425
Дополнительная информация .....	430
8.12. Управление биполярным шаговым двигателем .....	431
РЕШЕНИЕ .....	431
Обсуждение работы решения и возможных проблем.....	432
Дополнительная информация .....	435
8.13. Управление биполярным шаговым двигателем с использованием платы EasyDriver .....	435
ЗАДАЧА .....	435
РЕШЕНИЕ .....	435
Обсуждение работы решения и возможных проблем.....	438
8.14. Управление униполярным шаговым двигателем с помощью драйвера ULN2003А .....	439
ЗАДАЧА .....	439
РЕШЕНИЕ .....	439
Обсуждение работы решения и возможных проблем.....	441
Дополнительная информация .....	441
<b>Глава 9. Работа со звуком .....</b>	<b>443</b>
9.0. Введение .....	443
9.1. Воспроизведение звуков разной частоты.....	447
ЗАДАЧА .....	447
РЕШЕНИЕ .....	447
Дополнительная информация .....	450

9.2. Проигрывание простой мелодии .....	450
ЗАДАЧА .....	450
РЕШЕНИЕ .....	450
9.3. Генерирование несколько тонов одновременно.....	452
ЗАДАЧА .....	452
РЕШЕНИЕ .....	452
Обсуждение работы решения и возможных проблем.....	453
9.4. Генерирование звуков, не лишая себя возможности использовать ШИМ-сигнал.....	454
ЗАДАЧА .....	454
РЕШЕНИЕ .....	454
Обсуждение работы решения и возможных проблем.....	456
Дополнительная информация .....	457
9.5. Управление устройствами MIDI.....	457
ЗАДАЧА .....	457
РЕШЕНИЕ .....	457
Обсуждение работы решения и возможных проблем.....	459
Дополнительная информация .....	461
9.6. Создание синтезатора звуков .....	461
ЗАДАЧА .....	461
РЕШЕНИЕ .....	461
Обсуждение работы решения и возможных проблем.....	462
Дополнительная информация .....	463
9.7. Синтез звуков высокого качества.....	463
ЗАДАЧА .....	463
РЕШЕНИЕ .....	463
Обсуждение работы решения и возможных проблем.....	465
Дополнительная информация .....	466
<b>Глава 10. Дистанционное управление внешними устройствами.....</b>	<b>467</b>
10.0. Введение .....	467
10.1. Реагирование на команды ИК-пульта ДУ .....	468
ЗАДАЧА .....	468
РЕШЕНИЕ .....	468
Обсуждение работы решения и возможных проблем.....	470
Дополнительная информация .....	471
10.2. Декодирование ИК-сигналов управления пульта ДУ .....	472
ЗАДАЧА .....	472
РЕШЕНИЕ .....	472
Обсуждение работы решения и возможных проблем.....	475
Дополнительная информация .....	476
10.3. Имитация ИК-сигналов управления пульта ДУ .....	476
ЗАДАЧА .....	476
РЕШЕНИЕ .....	476
Обсуждение работы решения и возможных проблем.....	478
Дополнительная информация .....	479
10.4. Управление цифровой фотокамерой .....	479
ЗАДАЧА .....	479
РЕШЕНИЕ .....	480
Обсуждение работы решения и возможных проблем.....	482
Дополнительная информация .....	482
10.5. Управление электрическими устройствами с помощью модифицированного дистанционно управляемого выключателя .....	482
ЗАДАЧА .....	482
РЕШЕНИЕ .....	483

Обсуждение работы решения и возможных проблем.....	485
Дополнительная информация .....	486
<b>Глава 11. Работа с дисплеями .....</b>	<b>487</b>
11.0. Введение.....	487
11.1. Подключение и использование текстового ЖКД.....	488
ЗАДАЧА .....	488
РЕШЕНИЕ .....	488
Обсуждение работы решения и возможных проблем.....	491
Дополнительная информация .....	492
11.2. Форматирование выводимого текста.....	492
ЗАДАЧА .....	492
РЕШЕНИЕ .....	492
Обсуждение работы решения и возможных проблем.....	493
Дополнительная информация .....	495
11.3. Включение и выключение курсора и дисплея .....	495
ЗАДАЧА .....	495
РЕШЕНИЕ .....	495
Обсуждение работы решения и возможных проблем.....	496
11.4. Прокрутка текста.....	497
ЗАДАЧА .....	497
РЕШЕНИЕ .....	497
Обсуждение работы решения и возможных проблем.....	498
11.5. Отображения на экране ЖКД специальных символов.....	500
ЗАДАЧА .....	500
РЕШЕНИЕ .....	500
Обсуждение работы решения и возможных проблем.....	502
Дополнительная информация .....	503
11.6. Создание пользовательских символов.....	503
ЗАДАЧА .....	503
РЕШЕНИЕ .....	503
Обсуждение работы решения и возможных проблем.....	504
11.7. Отображение символов большего размера, чем стандартные.....	505
ЗАДАЧА .....	505
РЕШЕНИЕ .....	506
Обсуждение работы решения и возможных проблем.....	508
Дополнительная информация .....	509
11.8. Отображение символов меньшего размера, чем стандартные .....	509
ЗАДАЧА .....	509
РЕШЕНИЕ .....	509
Обсуждение работы решения и возможных проблем.....	510
11.9. Выбор графического жидкокристаллического дисплея.....	512
ЗАДАЧА .....	512
РЕШЕНИЕ .....	512
Обсуждение работы решения и возможных проблем.....	513
Дополнительная информация .....	514
11.10. Управление полноцветным жидкокристаллическим дисплеем .....	514
ЗАДАЧА .....	514
РЕШЕНИЕ .....	514
Обсуждение работы решения и возможных проблем.....	516
Дополнительная информация .....	518



11.11. Управление монохромным дисплеем OLED .....	518
ЗАДАЧА .....	518
РЕШЕНИЕ .....	518
Обсуждение работы решения и возможных проблем .....	521
Дополнительная информация .....	524
<b>Глава 12. Работа с временем и датами .....</b>	<b>525</b>
12.0. Введение .....	525
12.1. Использование функции <i>millis()</i> для определения длительности периода времени .....	525
ЗАДАЧА .....	525
РЕШЕНИЕ .....	525
Обсуждение работы решения и возможных проблем .....	526
Дополнительная информация .....	526
12.2. Создание пауз в скетче .....	526
ЗАДАЧА .....	526
РЕШЕНИЕ .....	527
Обсуждение работы решения и возможных проблем .....	527
Дополнительная информация .....	531
12.3. Точное измерение длительности импульса .....	532
ЗАДАЧА .....	532
РЕШЕНИЕ .....	532
Обсуждение работы решения и возможных проблем .....	533
Дополнительная информация .....	533
12.4. Использование платы Arduino в качестве часов .....	534
ЗАДАЧА .....	534
РЕШЕНИЕ .....	534
Обсуждение работы решения и возможных проблем .....	535
Дополнительная информация .....	543
12.5. Создание события для периодического вызова функции .....	543
ЗАДАЧА .....	543
РЕШЕНИЕ .....	543
Обсуждение работы решения и возможных проблем .....	545
12.6. Работа с часами реального времени .....	547
ЗАДАЧА .....	547
РЕШЕНИЕ .....	548
Обсуждение работы решения и возможных проблем .....	549
Дополнительная информация .....	552
<b>Глава 13. Протоколы связи I<sup>2</sup>C и SPI .....</b>	<b>553</b>
13.0. Введение .....	553
Интерфейс I <sup>2</sup> C .....	555
Использование устройств с напряжением питания 3,3 В совместно с платами с напряжением питания 5 В .....	556
Интерфейс SPI .....	558
Дополнительная информация .....	560
13.1. Подключение нескольких устройств I <sup>2</sup> C .....	560
ЗАДАЧА .....	560
РЕШЕНИЕ .....	560
Обсуждение работы решения и возможных проблем .....	562
Дополнительная информация .....	564
13.2. Подключение нескольких устройств SPI .....	564
ЗАДАЧА .....	564
РЕШЕНИЕ .....	564

Обсуждение работы решения и возможных проблем.....	566
Дополнительная информация .....	567
13.3. Работа с микросхемами на интерфейсе I <sup>2</sup> C.....	568
ЗАДАЧА .....	568
РЕШЕНИЕ .....	568
Обсуждение работы решения и возможных проблем.....	570
Дополнительная информация .....	572
13.4. Увеличение количества портов I <sup>2</sup> C .....	572
ЗАДАЧА .....	572
РЕШЕНИЕ .....	572
Обсуждение работы решения и возможных проблем.....	574
Дополнительная информация .....	576
13.5. Организация взаимодействия нескольких плат Arduino.....	576
ЗАДАЧА .....	576
РЕШЕНИЕ .....	577
Обсуждение работы решения и возможных проблем.....	579
Дополнительная информация .....	582
13.6. Использование возможностей акселерометра контроллера Wii Nunchuk.....	582
ЗАДАЧА .....	582
РЕШЕНИЕ .....	582
Обсуждение работы решения и возможных проблем.....	585
Дополнительная информация .....	588
<b>Глава 14. Простая беспроводная связь .....</b>	<b>589</b>
14.0. Введение .....	589
14.1. Обмен сообщениями с помощью недорогих радиомодулей .....	589
ЗАДАЧА .....	589
РЕШЕНИЕ .....	589
Обсуждение работы решения и возможных проблем.....	594
Дополнительная информация .....	598
14.2. Подключение плат Arduino по сети ZigBee (802.15.4).....	598
ЗАДАЧА .....	598
РЕШЕНИЕ .....	599
Обсуждение работы решения и возможных проблем.....	601
Конфигурирование радиомодуля XBee.....	602
Взаимодействие с платой Arduino .....	606
Дополнительная информация .....	606
14.3. Обмен сообщениями с конкретным радиомодулем XBee.....	606
ЗАДАЧА .....	606
РЕШЕНИЕ .....	607
Обсуждение работы решения и возможных проблем.....	608
Дополнительная информация .....	610
14.4. Обмен данными между радиомодулями XBee .....	610
ЗАДАЧА .....	610
РЕШЕНИЕ .....	610
Обсуждение работы решения и возможных проблем.....	611
Конфигурирование радиомодулей.....	611
Дополнительная информация .....	617
14.5. Активирование подключенного к радиомодулю XBee устройства.....	617
ЗАДАЧА .....	617
РЕШЕНИЕ .....	617

Обсуждение работы решения и возможных проблем.....	618
Радиомодули XBee Series 2 и XBee 3 .....	618
Радиомодули XBee Series 1 .....	621
Дополнительная информация .....	623
14.6. Взаимодействие с классическими устройствами Bluetooth.....	623
ЗАДАЧА .....	623
РЕШЕНИЕ .....	623
Обсуждение работы решения и возможных проблем.....	625
Дополнительная информация .....	626
14.7. Работа с радиомодулями Bluetooth LE .....	626
ЗАДАЧА .....	626
РЕШЕНИЕ .....	626
Обсуждение работы решения и возможных проблем.....	628
Дополнительная информация .....	629
<b>Глава 15. Сети Wi-Fi и Ethernet.....</b>	<b>631</b>
15.0. Введение .....	631
Среда Ethernet .....	632
Среда Wi-Fi.....	632
Протоколы TCP и IP .....	632
Локальные IP-адреса .....	632
Протокол HTTP .....	633
Язык разметки HTML .....	633
Потоковый парсинг Stream.....	633
Интерфейс API для веб-сайта.....	633
15.1. Подключение к сети Ethernet .....	634
ЗАДАЧА .....	634
РЕШЕНИЕ .....	634
Обсуждение работы решения и возможных проблем.....	635
Дополнительная информация .....	639
15.2. Получение IP-адреса устройства автоматически .....	639
ЗАДАЧА .....	639
РЕШЕНИЕ .....	639
Обсуждение работы решения и возможных проблем.....	640
15.3. Обмен простыми сообщениями по протоколу UDP.....	641
ЗАДАЧА .....	641
РЕШЕНИЕ .....	641
Обсуждение работы решения и возможных проблем.....	644
15.4. Использование платы Arduino со встроенным модулем Wi-Fi .....	650
ЗАДАЧА .....	650
РЕШЕНИЕ .....	650
Обсуждение работы решения и возможных проблем.....	653
15.5. Подключение к сети Wi-Fi с помощью недорогих модулей .....	654
ЗАДАЧА .....	654
РЕШЕНИЕ .....	654
Обсуждение работы решения и возможных проблем.....	658
Дополнительная информация .....	660
15.6. Извлечение данных из ответа веб-сервера.....	660
ЗАДАЧА .....	660
РЕШЕНИЕ .....	660
Обсуждение работы решения и возможных проблем.....	666
Дополнительная информация .....	667

15.7. Запрос данных у веб-сервера, использующего формат XML.....	667
ЗАДАЧА .....	667
РЕШЕНИЕ .....	667
Обсуждение работы решения и возможных проблем.....	669
15.8. Организация веб-сервера на платформе Arduino.....	670
ЗАДАЧА .....	670
РЕШЕНИЕ .....	670
Обсуждение работы решения и возможных проблем.....	676
15.9. Обработка входящих запросов от веб-клиентов.....	677
ЗАДАЧА .....	677
РЕШЕНИЕ .....	677
Обсуждение работы решения и возможных проблем.....	681
15.10. Обработка входящих запросов для конкретных страниц .....	682
ЗАДАЧА .....	682
РЕШЕНИЕ .....	682
Обсуждение работы решения и возможных проблем.....	685
15.11. Использование HTML для форматирования ответов веб-сервера.....	688
ЗАДАЧА .....	688
РЕШЕНИЕ .....	688
Обсуждение работы решения и возможных проблем.....	692
Дополнительная информация .....	692
15.12. Запрос данных посредством форм (метод POST) .....	693
ЗАДАЧА .....	693
РЕШЕНИЕ .....	693
Обсуждение работы решения и возможных проблем.....	696
15.13. Раздача веб-страниц, содержащих большие объемы данных .....	697
ЗАДАЧА .....	697
РЕШЕНИЕ .....	697
Обсуждение работы решения и возможных проблем.....	705
Дополнительная информация .....	706
15.14. Отправка сообщений в Twitter .....	706
ЗАДАЧА .....	706
РЕШЕНИЕ .....	706
Обсуждение работы решения и возможных проблем.....	709
Дополнительная информация .....	709
15.15. Организация обмена данными для Интернета вещей .....	709
ЗАДАЧА .....	709
РЕШЕНИЕ .....	709
Обсуждение работы решения и возможных проблем.....	710
Дополнительная информация .....	710
15.16. Публикация данных на брокере MQTT.....	710
ЗАДАЧА .....	710
РЕШЕНИЕ .....	710
Обсуждение работы решения и возможных проблем.....	712
Дополнительная информация .....	712
15.17. Подписка на данные брокера MQTT .....	712
ЗАДАЧА .....	712
РЕШЕНИЕ .....	712
Обсуждение работы решения и возможных проблем.....	714
15.18. Получение значения текущего времени от интернет-сервера времени.....	715
ЗАДАЧА .....	715
РЕШЕНИЕ .....	715

Обсуждение работы решения и возможных проблем.....	717
Дополнительная информация .....	721
<b>Глава 16. Использование, модифицирование и создание библиотек .....</b>	<b>723</b>
16.0. Введение .....	723
16.1. Использование встроенных библиотек .....	723
ЗАДАЧА .....	723
РЕШЕНИЕ .....	723
Обсуждение работы решения и возможных проблем.....	726
Дополнительная информация .....	726
16.2. Установка библиотек сторонних разработчиков.....	726
ЗАДАЧА .....	726
РЕШЕНИЕ .....	726
Обсуждение работы решения и возможных проблем.....	728
16.3. Модифицирование библиотеки.....	728
ЗАДАЧА .....	728
РЕШЕНИЕ .....	729
Обсуждение работы решения и возможных проблем.....	732
Дополнительная информация .....	733
16.4. Создание собственных библиотек .....	733
ЗАДАЧА .....	733
РЕШЕНИЕ .....	733
Обсуждение работы решения и возможных проблем.....	735
Дополнительная информация .....	740
16.5. Создание библиотеки, использующей другие библиотеки .....	740
ЗАДАЧА .....	740
РЕШЕНИЕ .....	740
Обсуждение работы решения и возможных проблем.....	744
Дополнительная информация .....	747
16.6. Обновление библиотек сторонних разработчиков для Arduino 1.0.....	747
ЗАДАЧА .....	747
РЕШЕНИЕ .....	747
Обсуждение работы решения и возможных проблем.....	748
<b>Глава 17. Продвинутое программирование и управления памятью.....</b>	<b>749</b>
17.0. Введение .....	749
Препроцессор .....	750
Дополнительная информация .....	751
17.1. Процесс сборки скетчей Arduino .....	751
ЗАДАЧА .....	751
РЕШЕНИЕ .....	751
Обсуждение работы решения и возможных проблем.....	751
Дополнительная информация .....	754
17.2. Определение объема свободной и занятой памяти RAM.....	754
ЗАДАЧА .....	754
РЕШЕНИЕ .....	755
Обсуждение работы решения и возможных проблем.....	756
Дополнительная информация .....	759
17.3. Использование программной флеш-памяти для записи и чтения числовых значений.....	759
ЗАДАЧА .....	759
РЕШЕНИЕ .....	759

Обсуждение работы решения и возможных проблем.....	761
Дополнительная информация .....	762
17.4. Использование программной флеш-памяти для записи и чтения строковых значений .....	762
ЗАДАЧА .....	762
РЕШЕНИЕ .....	762
Обсуждение работы решения и возможных проблем.....	764
Дополнительная информация .....	764
17.5. Использование вместо целых чисел ключевых слов <i>#define</i> и <i>const</i> .....	764
ЗАДАЧА .....	764
РЕШЕНИЕ .....	764
Обсуждение работы решения и возможных проблем.....	765
Дополнительная информация .....	766
17.6. Условное компилирование .....	766
ЗАДАЧА .....	766
РЕШЕНИЕ .....	766
Обсуждение работы решения и возможных проблем.....	767
Дополнительная информация .....	768
<b>Глава 18. Работа с аппаратными средствами микроконтроллера платы.....</b>	<b>769</b>
18.0. Введение .....	769
Регистры .....	770
Прерывания .....	770
Таймеры .....	771
Аналоговые и цифровые контакты.....	773
Дополнительная информация .....	773
18.1. Запись данных в память EEPROM.....	774
ЗАДАЧА .....	774
РЕШЕНИЕ .....	774
Обсуждение работы решения и возможных проблем.....	777
Дополнительная информация .....	778
18.2. Автоматическое реагирование на изменение состояния контакта .....	778
ЗАДАЧА .....	778
РЕШЕНИЕ .....	778
Обсуждение работы решения и возможных проблем.....	780
Дополнительная информация .....	781
18.3. Выполнение периодических действий .....	781
ЗАДАЧА .....	781
РЕШЕНИЕ .....	782
Обсуждение работы решения и возможных проблем.....	783
Дополнительная информация .....	784
18.4. Задание периода и длительности импульса .....	784
ЗАДАЧА .....	784
РЕШЕНИЕ .....	784
Обсуждение работы решения и возможных проблем.....	785
Дополнительная информация .....	786
18.5. Создание генератора импульсов .....	787
ЗАДАЧА .....	787
РЕШЕНИЕ .....	787
Обсуждение работы решения и возможных проблем.....	789
Дополнительная информация .....	790
18.6. Изменение частоты ШИМ-сигнала таймера .....	790
ЗАДАЧА .....	790
РЕШЕНИЕ .....	790

Обсуждение работы решения и возможных проблем.....	792
Дополнительная информация .....	793
18.7. Подсчет импульсов .....	793
ЗАДАЧА .....	793
РЕШЕНИЕ .....	793
Обсуждение работы решения и возможных проблем.....	794
Дополнительная информация .....	795
18.8. Измерение характеристик импульсов с более высокой точностью .....	795
ЗАДАЧА .....	795
РЕШЕНИЕ .....	796
Обсуждение работы решения и возможных проблем.....	798
Дополнительная информация .....	799
18.9. Оперативное измерение аналоговых значений.....	799
ЗАДАЧА .....	799
РЕШЕНИЕ .....	799
Обсуждение работы решения и возможных проблем.....	801
Дополнительная информация .....	801
18.10. Понижение энергопотребления приложения.....	801
ЗАДАЧА .....	801
РЕШЕНИЕ .....	801
Обсуждение работы решения и возможных проблем.....	802
Дополнительная информация .....	804
18.11. Быстрая установка уровней на цифровых контактах.....	804
ЗАДАЧА .....	804
РЕШЕНИЕ .....	804
Обсуждение работы решения и возможных проблем.....	807
18.12. Загрузка скетчей с помощью программатора .....	808
ЗАДАЧА .....	808
РЕШЕНИЕ .....	808
Обсуждение работы решения и возможных проблем.....	809
Дополнительная информация .....	809
18.13. Обновление или замена загрузчика Arduino.....	810
ЗАДАЧА .....	810
РЕШЕНИЕ .....	810
Обсуждение работы решения и возможных проблем.....	810
Дополнительная информация .....	811
18.14. Перемещение курсора мыши компьютера.....	811
ЗАДАЧА .....	811
РЕШЕНИЕ .....	811
Обсуждение работы решения и возможных проблем.....	813
Дополнительная информация .....	813

## **Приложение 1. Электронные компоненты..... 815**

Конденсатор .....	815
Диод.....	816
Интегральные схемы.....	817
Цифровая клавиатура.....	817
Светодиоды.....	817
Электродвигатели постоянного тока.....	817
Оптрон.....	817
Фоторезисторы .....	818
Пьезоэлектрический зуммер .....	818

Потенциометр.....	818
Реле.....	818
Резистор .....	818
Соленоид.....	818
Динамик .....	819
Шаговый электродвигатель.....	819
Переключатель .....	819
Транзистор.....	819
Дополнительная информация .....	819

## **Приложение 2. Работа с принципиальными схемами и справочными**

<b>листочками .....</b>	<b>821</b>
Принципиальные схемы .....	821
Справочные листки.....	824
Справочные листки и выбор транзисторов для использования в схемах Arduino.....	824

## **Приложение 3. Сборка схем .....**

<b>827</b>	
Работа с безопасной макетной платой .....	827
Использование внешних источников питания .....	828
Использование развязывающих конденсаторов.....	829
Использование демпферных диодов с индуктивными нагрузками .....	830
Работа с напряжением электросети .....	830

## **Приложение 4. Советы по диагностированию программных неполадок .....**

<b>831</b>	
Код, который не компилируется.....	831
Код компилируется, но не работает, как ожидалось.....	833

## **Приложение 5. Советы по диагностированию аппаратных неполадок .....**

<b>835</b>	
Ничего не помогло? .....	836

## **Приложение 6. Цифровые и аналоговые контакты .....**

## **Приложение 7. Коды ASCII для стандартного и расширенного наборов**

<b>символов.....</b>	<b>843</b>
----------------------	------------

<b>Предметный указатель.....</b>	<b>853</b>
----------------------------------	------------



---

# Начало работы

## 1.0. Введение

Платформа Arduino была разработана для того, чтобы начинающим пользователям, не обладающим опытом в области программирования или электроники, было легко работать с ней. С помощью платформы Arduino можно создавать устройства, которые будут реагировать на свет, звук, прикосновение или движение, и управлять ими. В частности, ее можно использовать для создания широкого диапазона устройств, включая музыкальные инструменты, роботов, световые арт-объекты, игры, интерактивную мебель и даже интерактивную одежду.

Платформа Arduino применяется во многих обучающих программах во всем мире, в особенности дизайнерами и творческими работниками, которые хотят иметь возможность быстро и легко создавать прототипы своих разработок, не требующие от них глубокого понимания технических подробностей их реализации. Поскольку платформа Arduino предназначена для использования людьми, не обладающими большим техническим опытом, ее программная часть содержит большой объем кода примеров, демонстрирующих использование различных возможностей платы Arduino.

Хотя с платформой Arduino легко работать, уровень сложности аппаратной части, лежащей в ее основе, тот же самый, что и у аппаратного обеспечения, используемого для создания промышленных встроенных устройств. Платформа Arduino привлекательна и для пользователей, которые уже работают с микроконтроллерами, — из-за предоставляемых ею возможностей быстрой разработки, позволяющих намного ускорить процесс воплощения идеи в рабочий прототип изделия.

Платформа Arduino славится своей аппаратной частью, но для программирования аппаратной составляющей требуются программные средства. Поэтому мы и называем оба эти компонента — и аппаратную, и программную составляющие платформы — общим названием Arduino. Оригинальные и совместимые платы Arduino (аппаратная составляющая) не стоят больших денег, а при желании плату Arduino — благодаря открытости аппаратной составляющей — можно собрать и самому. Программное обеспечение для Arduino вообще распространяется бесплатно, имеет открытый код и совместимо с основными операционными системами. Комбинация всех этих свойств платформы позволяет использовать ее для создания проектов, которые могут взаимодействовать с окружающим миром и управлять объектами в нем.

Кроме этого, существует активное и всегда готовое прийти на помощь сообщество Arduino, которое доступно во всем мире через форумы Arduino (<https://forum.arduino.cc>), учебные пособия (<https://oreil.ly/eptlu>) и центр проектов Project Hub (<https://oreil.ly/1aGpz>). На этих веб-сайтах можно найти обучающие ресурсы, примеры разработки проектов и решения задач, которые могут помочь вам в реализации своих собственных проектов.

## Скетчи и программное обеспечение Arduino

Прикладные программы, которые на жаргоне Arduino называются *скетчами*, создаются на компьютере в интегрированной среде разработки (IDE) Arduino. Среда IDE позволяет создавать и редактировать код программ, а затем преобразовывать этот код в машинные инструкции для исполнения на плате Arduino. Среда IDE также используется для передачи этих машинных инструкций в виде скомпилированного кода на плату Arduino. Этот процесс называется *загрузкой скетча*.



Может быть, вы привыкли, что исходный код программы называется «программа» или просто «код». В сообществе Arduino исходный код, содержащий компьютерные инструкции для управления функциональностями Arduino, называется *скетчем*. Слово *скетч* будет использоваться в этой книге для обозначения исходного кода прикладных программ для Arduino.

Примеры из этой главы помогут вам приступить к работе, поясняя, как установить и настроить среду разработки Arduino и как скомпилировать и запустить на исполнение скетч примера.

Скетч Blink для мигания светодиодом, который установлен по умолчанию на большинстве оригинальных и совместимых плат Arduino, используется в этой главе в качестве эталонного примера. Но последний пример этой главы, кроме мигания светодиодом, обладает большими способностями. В частности, с помощью дополнительных аппаратных средств, подключенных к плате Arduino, он может воспроизводить звуки и принимать входные данные.

Структура скетча Arduino рассматривается в *главе 2*, где и дается введение в программирование.



Если вы уже обладаете базовыми знаниями по Arduino, вы вполне можете пропустить этот материал и перейти к материалу, который вам еще не знаком. Но если это ваше первое знакомство с Arduino, то терпеливое изучение этих начальных примеров поможет вам разрабатывать свои скетчи с большей уверенностью в дальнейшем.

## Аппаратная часть платформы Arduino

Код написанной программы исполняется на плате Arduino. Эта плата может реагировать только на электрические сигналы и управлять только ими, поэтому для придания ей возможности взаимодействия с внешним миром к ней подключаются различные дополнительные аппаратные устройства. Такими устройствами могут быть датчики, которые преобразовывают различные внешние воздействия в электриче-

ские сигналы, подаваемые затем на плату Arduino. А подключенные к плате исполнительные устройства преобразовывают генерируемые платой Arduino электрические сигналы в какой-либо тип физического действия. Примером датчиков могут служить переключатели, акселерометры, ультразвуковые дальнометры и т. п., а исполнительных устройств — светодиоды, динамики, электродвигатели, дисплеи и т. п.

На рынке предлагается большой выбор оригинальных плат Arduino и еще больший выбор совместимых плат, изготавливаемых как компаниями, так и отдельными членами сообщества Arduino. Установленные в платах Arduino микроконтроллеры используются, кроме самих плат, также в широком диапазоне других устройств — от 3D-принтеров до роботов. Некоторые из этих совместимых с Arduino плат и устройств также совместимы с другими средами программирования — такими как MicroPython или CircuitPython.

Самые популярные платы Arduino оснащены разъемом USB, который служит одновременно как для подачи питания на плату, так и для взаимодействия платы с компьютером, — в частности, для загрузки в плату скетчей. На рис. 1.1 показана базовая плата Arduino Uno, которая используется большинством начинающих пользователей. На плате установлен 8-разрядный микропроцессор ATmega328P, несущий на борту 2 Кбайт памяти SRAM (статическая память с произвольным доступом, служащая для хранения переменных программы) и 32 Кбайт флеш-памяти — для хранения скетчей. Работает этот процессор на частоте 16 МГц. Для обеспечения интерфейса USB на плате Arduino Uno установлен дополнительный микроконтроллер.

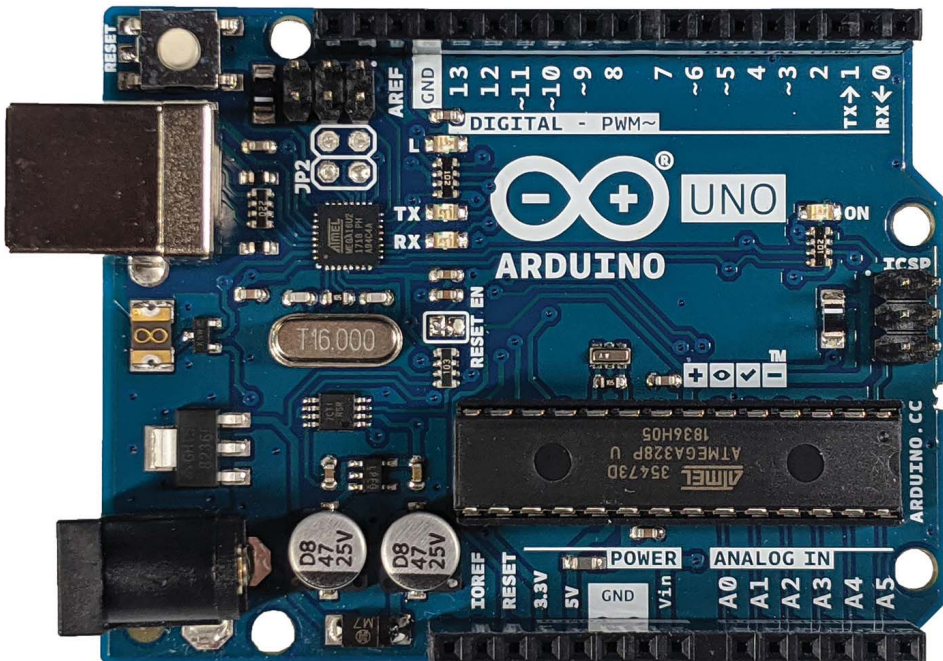


Рис. 1.1. Базовая плата Arduino Uno

Плата Arduino Leonardo имеет такой же *форм-фактор* (размер и организацию платы и ее контактов), что и плата Arduino Uno, но в ней установлен другой микроконтроллер — ATmega32U4. Этот микроконтроллер используется как для исполнения скетчей, так и для обеспечения интерфейса USB. Плата Arduino Leonardo чуть дешевле, чем плата Arduino Uno, но также поддерживает несколько интересных возможностей — таких как эмулирование различных USB-устройств, например клавиатуры и мыши. Совместимые с Arduino платы Teensy и Teensy++ компании PJRC (<http://www.pjrc.com/teensy>) также могут эмулировать устройства USB.

Плата Arduino Zero имеет похожее размещение контактов и даже более быстрый микроконтроллер. Но, в отличие от плат Uno и Leonardo, на контакты этой платы нельзя подавать сигналы напряжением выше чем 3,3 В. Плата Arduino Zero оснащена 32-разрядным микроконтроллером, который имеет 32 Кбайт памяти RAM и 256 Кбайт флеш-памяти и работает на частоте 48 МГц.

Плата Metro M0 Express компании Adafruit и плата RedBoard Turbo компании SparkFun имеют такой же *форм-фактор*, что и плата Arduino Zero, и также совместимы с разными средами разработки, включая среду Arduino IDE и CircuitPython.

---

### Arduino и USB

---

Плата Arduino Uno оснащена дополнительным микроконтроллером для обеспечения интерфейса USB. Этот микроконтроллер в версии для поверхностного монтажа (ATmega16U2 или ATmega8U2 — в более старых версиях платы Uno) установлены на плате возле разъема USB. А плата Arduino Leonardo имеет только один микроконтроллер ATmega32U4, который отвечает как за исполнение кода скетчей, так и за интерфейс USB. Плату Leonardo можно перепрограммировать, чтобы она эмулировала другие устройства USB (см. *разд. 18.14*).

Более старые платы Arduino и некоторые совместимые с Arduino платы используют микросхему компании FTDI, которая реализует на аппаратном уровне подключение USB через последовательный порт компьютера. Некоторые дешевые клоны платы Arduino, которые предлагаются на площадках eBay или Amazon, могут использовать другую микросхему, например CH340, которая реализует такую же функцию. Для таких плат может потребоваться установить соответствующий драйвер.

Также существует класс совместимых с Arduino плат с поддержкой USB-интерфейса без использования аппаратных средств. Вместо этого в таких платах задействован программный подход, когда исполняющаяся на плате программа управляет контактами ввода/вывода для обмена сигналами USB. Такие платы, в число которых входит популярная оригинальная плата Trinket компании Adafruit, могут испытывать проблемы с работой при подключении к современным компьютерам, хотя с более старыми компьютерами могут работать должным образом. (Компания Adafruit выпустила плату Trinket M0, оснащенную аппаратной поддержкой USB, и которая, как бонус, намного быстрее, чем ее предшественница.)

Наконец, встречаются совместимые с Arduino платы, вообще не поддерживающие USB-подключение. Вместо этого они оснащены только контактами последовательного интерфейса, и для подключения к компьютеру через эти контакты требуют специального адаптера. Дополнительная информация по некоторым таким адаптерам приводится в *разд. «Аппаратные средства последовательной связи» главы 4*.

---

Если вы хотите иметь плату, пригодную для изучения возможностей Arduino, и на которой можно исполнить большинство скетчей из этой книги, плата Uno — прекрасный выбор для этих целей. Но если вам нужна плата, обладающая более высоким быстродействием, чем Uno, но при этом имеющая такой же *форм-фактор*, тогда обратите внимание на плату Zero или подобные ей платы — например, на плату

Metro M0 Express или RedBoard Turbo. Платы ряда MKR и Nano 33 также обладают отличным быстродействием, но имеют меньший форм-фактор, чем плата Uno.



### Будьте осторожны с некоторыми платами, имеющими напряжение рабочего питания 3,3 В

Многие новые платы используют рабочее напряжение величиной 3,3 В, а не 5 В, как более старые платы, такие как Uno. Такие платы могут быть безвозвратно повреждены, если на какой-либо из их контактов ввода/вывода подать сигнал напряжением 5 В даже на малую долю секунды. Поэтому, если существует возможность подачи на контакты ввода/вывода такой платы сигнала напряжением выше чем 3,3 В, проверьте в документации на плату, способна ли она выдержать такое напряжение. В большинстве плат с рабочим напряжением 3,3 В на входной разъем рабочего напряжения подается напряжение величиной 5 В (например, через порт USB), но это напряжение понижается преобразователем напряжения до 3,3 В, прежде чем подаваться на электронные компоненты платы. В результате этого платы, чьи контакты ввода/вывода не допускают сигналов напряжением 5 В, весьма часто имеют контакт с выходным рабочим питанием 5 В.

Существуют платы Arduino и других форм-факторов, а это значит, что организация контактов на таких платах иная, чем у платы Uno, и они не совместимы с шилдами, предназначенными для работы с платой Uno. Плата MKR1010 — одна из таких плат Arduino с иным форм-фактором, намного меньшим, чем у платы Uno. Контакты ввода/вывода этой платы могут выдерживать сигналы напряжением только 3,3 В и в ней, так же как и в плате Zero, используется процессор ARM. Но плата MKR1010 оснащена встроенной поддержкой Wi-Fi и схемой для работы от литий-полимерной батареи и зарядки этой батареи. Хотя платы семейства MKR несовместимы с шилдами, предназначенными для платы Uno, для них предлагается свой выбор расширительных плат, называемых *носителями* (carriers).

### Расширение возможностей Arduino с помощью шилдов

Функциональность плат Arduino можно расширить с помощью расширительных модулей, называемых *шилдами*. Эти модули устанавливаются поверх платы Arduino, при этом их штыревые разъемы вставляются в гнездовые разъемы платы Arduino. Для разных моделей плат Arduino и некоторых совместимых с Arduino плат могут предлагаться расширительные модули, похожие на шилды, но несовместимые с ними. Это объясняется использованием в этих платах иного форм-фактора, чем в самой популярной плате Arduino Uno. Например, плата Arduino MKR имеет меньшие физические размеры, чем плата Uno. Платы расширения для плат MKR также называются шилдами, хотя их форм-фактор несовместим с платой Uno. Компания Adafruit предлагает для своих плат громадный выбор плат расширения Featherwing разработчика линейки Feather, которые совместимы с программными средствами разработки Arduino. Однако платы расширения Featherwing несовместимы с платами Arduino иных форм-факторов — такими как платы Uno и MKR.

На рынке предлагаются платы размером с почтовую марку — например, плата Trinket M0 компании Adafruit. Платы же большего размера предоставляют более богатый выбор опций подключения и оснащаются более мощными процессорами — например, платы Arduino Mega и Arduino Due. Также имеются платы для специальных вариантов использования — например, носимая плата Arduino LilyPad, беспроводная плата Arduino Nano 33 IoT и встраиваемая плата Arduino Nano Every (для автономных приложений, которые часто питаются от батареи).

Многие сторонние производители также предлагают свои платы, совместимые с Arduino:

- ◆ несмонтированные платы — это наборы компонентов платы Arduino, включая саму печатную плату, которые пользователь должен собрать самостоятельно. Компания Modern Device (<https://oreil.ly/aBakM>) предлагает такие наборы с поддержкой интерфейса USB и без таковой, а компания Educato (<https://oreil.ly/bY5YZ>) — наборы, совместимые с шилдами;
- ◆ платы компании Adafruit Industries (<http://www.adafruit.com>) — компания Adafruit предлагает огромный выбор как оригинальных, так совместимых плат Arduino и вспомогательных модулей и компонентов;
- ◆ платы компании SparkFun (<https://oreil.ly/rr5Ry>) — компания SparkFun предлагает большой выбор модулей и компонентов как для плат Arduino, так и для совместимых плат;
- ◆ платы компании Seeed Studio (<http://www.seeedstudio.com>) — компания Seeed Studio предлагает как оригинальные, так и совместимые платы Arduino, а также множество модулей и компонентов к этим платам. У них также можно приобрести гибкую расширительную систему Grove (<https://oreil.ly/pSKC8>) для плат Arduino и других автономных плат, предлагающую варианты модульных разъемов для подключения датчиков и приводов;
- ◆ платы Teensy and Teensy++ (<http://www.pjrc.com/teensy>) — эти крошечные, но в высшей степени многоцелевые платы, предлагаются компанией PJRC.

Исчерпывающий список совместимых с Arduino плат можно найти в Википедии (<https://oreil.ly/uyFoP>). А на веб-сайте Arduino (<https://oreil.ly/yay5b>) приводится обзор плат Arduino.

## 1.1. Установка интегрированной среды разработки Arduino IDE

### ЗАДАЧА

Установить на свой компьютер интегрированную среду разработки Arduino IDE.

### РЕШЕНИЕ

Загрузите установочный пакет интегрированной среды разработки Arduino IDE со страницы загрузок веб-сайта Arduino (<http://arduino.cc/download>) для своей операционной системы — Windows, macOS или Linux. Далее приводятся инструкции для установки среды на каждую из этих платформ.

#### Установка среды Arduino IDE на Windows

Если вы работаете на компьютере под Windows 10, среду разработки Arduino IDE установить на него без прав администратора можно, используя магазин Microsoft

Store. Но для более ранних версий Windows запуск загруженного установщика среды возможен только с правами администратора. Однако вместо установочного пакета можно загрузить ZIP-архив среды и распаковать его в любую папку, для которой у вас есть права записи.

В результате распаковки вы получите папку с названием Arduino-`<nn>`, где `<nn>` означает номер версии загруженной среды разработки Arduino IDE. Эта папка содержит исполняемый файл `Arduino.exe` и прочие файлы и папки среды.



При первом запуске среды Arduino IDE может открыться диалоговое окно с сообщением, что брандмауэр Защитника Windows заблокировал некоторые возможности этого приложения, указывая на файл `javaw.exe` как на причину такого предупреждения. То, что предупреждение вызывается программой `javaw.exe`, а не программой `Arduino.exe`, объясняется тем, что среда разработки Arduino IDE основана на платформе Java. Приложение `javaw.exe` используется платами, поддерживающими возможность загрузки в них скетчей по локальной сети. Если вы планируете работать с такими платами, вам нужно будет в окне предупреждения разрешить этому приложению доступ к локальной сети.

При подключении платы Arduino к компьютеру система автоматически ассоциирует установленные в ней драйверы с платой (это может занять некоторое время). Если же этот процесс не завершится успехом, или если вы установили среду Arduino IDE из ZIP-архива, вам придется установить драйверы для своей платы вручную. Для этого посетите веб-страницу *Getting Started with Arduino products* (<https://oreil.ly/ELrle>), в правой панели этой страницы щелкните на ссылке со своей платой и установите драйверы, следуя предлагаемым инструкциям.

Если вы используете более старую плату (т. е. любую плату с драйверами FTDI) и ваш компьютер подключен к Интернету, выполнение поиска драйверов для платы можно предоставить Windows, и они установятся автоматически. Если подключение к Интернету отсутствует, или если используется операционная система Windows XP, место расположения драйверов надо будет указать вручную. Для этого с помощью окна навигации по файловой системе выберите папку `drivers\FTDI USB Drivers` в папке, в которую вы распаковали ZIP-архив. После установки драйвера платы откроется диалоговое окно **Обнаружено новое оборудование**, сообщая об обнаружении нового последовательного порта. Снова выполните описанный ранее процесс установки драйвера.



Возможно, вам придется выполнить процесс установки драйверов дважды, чтобы среда Arduino IDE могла взаимодействовать с платой.

## Установка среды Arduino IDE на macOS

Для компьютеров под управлением macOS среда разработки Arduino IDE загружается в виде ZIP-архива. Если загруженный архив не распакуется автоматически, перейдите в папку, в которую он был загружен, и щелкните на нем двойным щелчком мыши, чтобы его распаковать. Переместите распакованную папку среды в какую-либо удобную папку — например, в папку Приложения, где ей самое место.

Запустите среду на исполнение, щелкнув двойным щелчком мыши на ее значке. Откроется начальный экран среды, после которого откроется главное окно программы.

Для современных плат Arduino — таких как Uno, установка дополнительных драйверов не требуется. При первом подключении платы к компьютеру может открыться диалоговое окно с сообщением, что обнаружен новый сетевой порт. Его можно закрыть, не предпринимая никаких действий. Для более старых плат может потребоваться установить драйверы FTDI, которые можно загрузить на веб-сайте этой компании ([https://oreil.ly/w7\\_YM](https://oreil.ly/w7_YM)).

## Установка среды Arduino IDE на Linux

Версии среды разработки Arduino IDE для Linux все чаще включаются разработчиками в пакет установочного дистрибутива этой операционной системы. Но обычно это не самые последние версии среды, поэтому лучше всего загрузить ее текущую версию с веб-сайта Arduino по адресу: <http://arduino.cc/download>. Среда Arduino IDE для Linux доступна как для 32- так и для 64-разрядных версий системы, предлагается также версия и для процессоров ARM, которую можно установить на одноплатные компьютеры Raspberry Pi и другие компьютеры с процессорами линейки ARM и операционной системой Linux. Большинство дистрибутивов используют стандартный предустановленный драйвер, а также обычно поддерживают интерфейс FTDI. Подробные инструкции по установке среды Arduino IDE на компьютеры под Linux вы найдете на веб-сайте Arduino по адресу: <https://www.arduino.cc/en/guide/linux>. Вкратце: вам надо будет запустить на исполнение сценарий установки, а также может потребоваться дать разрешение своей учетной записи на доступ к последовательному порту.

\* \* \*

Для совместимых с Arduino плат сторонних производителей может потребоваться установить файлы поддержки, используя для этого Менеджер плат (см. *разд. 1.7*). При возникновении необходимости в процессе установки среды Arduino IDE выполнить какие-либо дополнительные шаги для конкретной платы, обратитесь за информацией к документации для этой платы.

Завершив установку среды Arduino IDE, запустите ее на исполнение, щелкнув на значке программы двойным щелчком, — в результате должно открыться окно начальной заставки (рис. 1.2).

За начальным экраном-заставкой следует главное окно программы, показанное на рис. 1.3. Для открытия главного окна может потребоваться некоторое время, чтобы загрузился требуемый код, поэтому запаситесь терпением. Значок для запуска среды Arduino IDE должен присутствовать в меню **Пуск | Все программы** Windows и в папке Приложения macOS, а также, возможно, на рабочем столе компьютера. На системах под Linux среда Arduino IDE запускается из оболочки терминала.



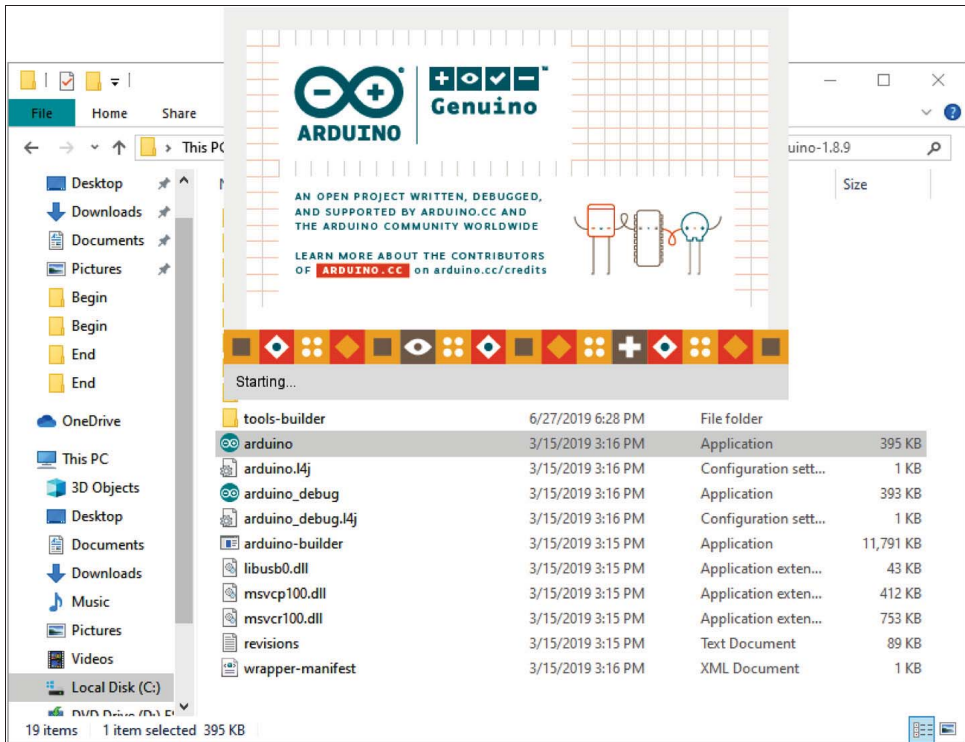


Рис. 1.2. Окно заставки при запуске среды Arduino IDE (Arduino 1.8.9 на Windows 10)

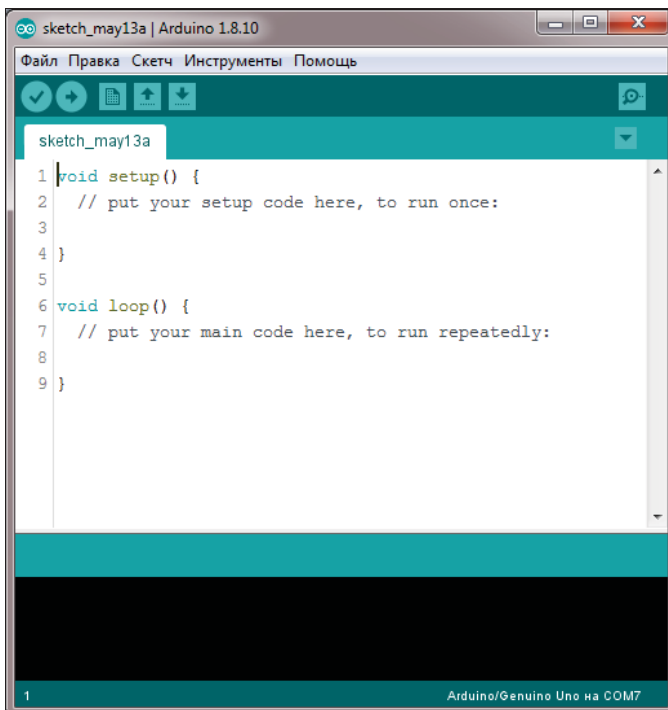


Рис. 1.3. Главное окно среды Arduino IDE (Arduino 1.8.9 на macOS)

## Возможные проблемы установки

Если среда Arduino IDE не запускается, инструкции по диагностированию возможных причин этого можно найти в разделе устранения неполадок веб-сайта Arduino по адресу: <https://oreil.ly/sllus>.

## Дополнительная информация

На веб-сайте Arduino приводятся подробные инструкции по установке среды Arduino IDE для Windows (<https://oreil.ly/ufBhk>), macOS (<https://oreil.ly/GL4uk>) и Linux (<https://oreil.ly/TW6si>).

Кроме стандартной среды разработки Arduino IDE также может использоваться среда Arduino Pro, ориентированная на профессиональных пользователей. Но на момент подготовки книги эта среда была только в начальной степени готовности. Дополнительную информацию по среде Arduino Pro можно найти на веб-странице ее репозитория GitHub (<https://oreil.ly/tKGNk>).

Инструмент командной строки Arduino CLI предназначен для компилирования и загрузки скетчей. Его также можно использовать вместо Менеджера библиотек и Менеджера плат. Дополнительную информацию по этому инструменту вы найдете на веб-странице его репозитория GitHub (<https://oreil.ly/yaOph>).

Наконец, доступна также и онлайн-среда разработки Arduino Create. Чтобы работать с этой средой, необходимо создать на веб-сайте Arduino учетную запись и установить на свой компьютер программный подключаемый модуль, который позволяет веб-сайту взаимодействовать с подключенной к компьютеру платой для загрузки в нее кода. Разрабатываемые скетчи сохраняются в облачном хранилище, и доступ к ним можно предоставлять другим разработчикам. На момент подготовки этой книги среда Arduino Create представляет собой достаточно новый сервис, все еще находящийся в стадии развития. Впрочем, если вы хотите разрабатывать скетчи Arduino, не устанавливая среду разработки на свой компьютер, среда Arduino Create (<https://create.arduino.cc>) сможет вам в этом помочь.

Для пользователей Chromebook в среде Arduino Create предлагается приложение Chrome, доступ к которому требует подписки стоимостью \$1 в месяц. Впрочем, это приложение предусматривает также и временный бесплатный доступ, чтобы можно было с ним ознакомиться. Альтернативу среде Arduino Create для компилирования и загрузки кода с устройств Chromebook предлагает и онлайн-среда разработки Codebender. Эта среда также поддерживает некоторые совместимые с Arduino платы сторонних производителей. Бесплатная версия среды имеет ограниченные возможности, которые снимаются при оформлении одного из типа подписок — для класса, школы или школьного округа. Дополнительную информацию о среде разработки Codebender можно найти на веб-сайте этого сервиса (<https://edu.codebender.cc>).

## 1.2. Подготовка платы Arduino к работе

### ЗАДАЧА

Подать питание на плату Arduino и удостовериться, что она работает должным образом.

### РЕШЕНИЕ

Подключите плату к порту USB компьютера и проверьте, загорелся ли на ней светодиодный индикатор питания (рис. 1.4). Большинство плат Arduino оснащены таким индикатором, который загорается при подаче питания на плату.

Кроме этого, на плате должен начать мигать другой светодиодный индикатор. Большинство плат Arduino при изготовлении программируются простым скетчем, который мигает этим светодиодом, осуществляя тем самым простую проверку работоспособности платы.

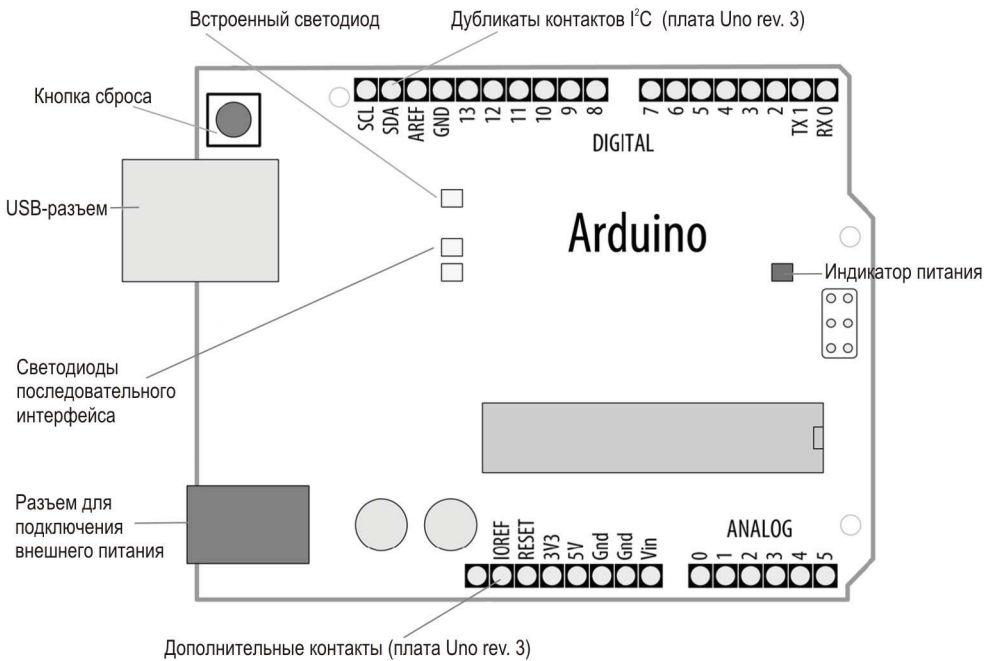


Рис. 1.4. Схема базовой платы Arduino Uno (Rev. 3)



На современных версиях платы Arduino Uno присутствуют несколько контактов, которых нет в старых версиях платы. Соответствующие контакты могут отсутствовать и в некоторых старых шилдах Arduino. К счастью, это обычно не сказывается на использовании старых шилдов, и большинство из них будут работать с новыми версиями платы так же, как и со старыми (но результаты могут быть разными для разных шилдов).

Дополнительный контакт IOREF в новых версиях платы служит для предоставления шилдам аналогового эталонного напряжения, чтобы входные аналоговые значения можно было соотнести с напряжением источника питания. А контакты SCL и SDA добавлены с целью обеспечить постоянное расположение контактов этих сигналов для устройств I<sup>2</sup>C. Вследствие разных конфигураций микроконтроллеров на некоторых более старых версиях плат (например, Mega) контакты I<sup>2</sup>C могли располагаться в разных местах и для некоторых шилдов нужно было прибегать к обходным маневрам — например, подключать контакты I<sup>2</sup>C шилда к соответствующим контактам платы с помощью проволочных перемычек. Теперь же шилды, разработанные под новую организацию контактов, должны работать со всеми платами с соответствующими дополнительными контактами. В настоящее время дополнительный контакт рядом с контактом IOREF не используется, но может быть задействован для реализации в будущем какой-либо новой функциональности, не требующей снова изменять организацию контактов.

## Обсуждение возможных проблем

Если при подключения платы к компьютеру ее индикатор питания не загорается, это, скорее всего, потому, что на плату не подается питание. Попробуйте подключить плату к другому разъему USB компьютера или с помощью другого кабеля USB.

Мигающий светодиод управляется исполняющимся на плате скетчем Blink, который загружается в плату при ее изготовлении. Мигание означает, что этот скетч выполняется должным образом, что, в свою очередь, свидетельствует о правильной работе платы. Если светодиод индикатора питания горит, но светодиод (обычно обозначенный буквой L) не мигает, причина этого может заключаться в том, что скетч Blink не был в плату загружен. Следуя инструкциям из *разд. 1.3*, загрузите в плату этот скетч, чтобы удостовериться в работоспособности платы. На платах иных, чем Uno, встроенный светодиод может отсутствовать — сверьтесь с документацией на свою плату, чтобы узнать, оснащена ли она таким светодиодом.

На платах Leonardo и платах класса Zero (Arduino Zero, Adafruit Metro M0, SparkFun RedBoard Turbo) гнездовые разъемы расположены так же, как и на плате Uno, позволяя подключение к ним стандартных шилдов. Но в прочих моментах эти платы значительно отличаются от платы Uno. Подобно плате Uno, плата Leonardo оснащена 8-разрядным микроконтроллером, но поскольку она не имеет отдельного микроконтроллера для обеспечения интерфейса USB, программы в нее можно загружать только в течение некоторого периода времени сразу же после сброса. Период времени, в течение которого плата Leonardo ожидает загрузки скетча, обозначается миганием встроенного светодиода платы. Плата Leonardo может работать с логическими сигналами напряжением 5 В.

В плате Zero используется 32-разрядный микропроцессор ARM, имеющий большой объем памяти для хранения и исполнения скетчей. Плата также оснащена встроенным цифроаналоговым преобразователем (ЦАП), выходной сигнал которого подается в виде варьирующегося напряжения на отдельный контакт ввода/вывода платы. С помощью ЦАП можно генерировать звуковые сигналы намного лучшего качества, чем посредством платы Uno. Плата Zero не поддерживает работу с логи-

ческими сигналами напряжением 5 В, так же как и подобные платы компании Adafruit (Metro M0 Express) и компании SparkFun (RedBoard Turbo).

Плата Arduino MKR1010 оснащена таким же микропроцессором, что и плата Zero, и также не поддерживает работу с логическими сигналами напряжением 5 В. При этом она имеет еще и меньший форм-фактор, т. е. несовместима с шилдами, предназначенными для плат форм-фактора Uno. Эта плата также поддерживает Wi-Fi, что позволяет подключать ее к Интернету через сеть Wi-Fi.

Все 32-разрядные платы имеют большее количество контактов, поддерживающих прерывания, чем большинство 8-разрядных плат. Прерывания полезны в приложениях, которые должны быстро определять изменения уровней сигнала (см. *разд 18.2*). Единственным исключением из этого является плата Arduino Uno WiFi Rev2, которая поддерживает прерывания на всех своих цифровых контактах ввода/вывода.

## Дополнительная информация

На веб-сайте Arduino приводятся подробные инструкции по установке среды Arduino IDE для Windows (<https://oreil.ly/ufBhk>), macOS (<https://oreil.ly/GL4uk>) и Linux (<https://oreil.ly/TW6si>). Инструкции для конкретной платы можно найти на странице Arduino Guide (<https://oreil.ly/MHLdN>).

На веб-сайте Arduino (<https://oreil.ly/IYeR0>) также приводится руководство по диагностированию проблем.

## 1.3. Создание скетча в среде разработки Arduino IDE

### ЗАДАЧА

Ознакомиться с процессом компиляции скетчей в среде Arduino IDE и разобраться с сообщениями об ошибках и статусными сообщениями.

### РЕШЕНИЕ

Исходный код программы Arduino называется *скетчем*. Процесс преобразования исходного кода скетча в форму, которая может исполняться платой Arduino, называется *компиляцией*. Для создания скетчей, открытия и модифицирования скетчей, которые определяют поведение платы Arduino, используется среда разработки Arduino IDE. Все указанные действия можно выполнять с помощью кнопок, расположенных в верхней части главного окна среды Arduino IDE (рис. 1.5). Впрочем, вы можете также воспользоваться меню (рис. 1.6) или соответствующими клавишами быстрого вызова.

Для ввода и редактирования кода скетча предназначена область редактора, которая занимает большую часть главного окна среды Arduino IDE. Редактор поддерживает распространенные клавиши быстрого вызова для редактирования текста:



Рис. 1.5. Элементы панели инструментов и окна среды Arduino IDE

- ◆  $\langle \text{Ctrl} \rangle + \langle \text{F} \rangle$  ( $\langle \text{Command} \rangle + \langle \text{F} \rangle$  на macOS) — для поиска текста;
- ◆  $\langle \text{Ctrl} \rangle + \langle \text{Z} \rangle$  ( $\langle \text{Command} \rangle + \langle \text{Z} \rangle$  на macOS) — для отмены последней операции;
- ◆  $\langle \text{Ctrl} \rangle + \langle \text{C} \rangle$  ( $\langle \text{Command} \rangle + \langle \text{C} \rangle$  на macOS) — для копирования выделенного текста в буфер обмена;
- ◆  $\langle \text{Ctrl} \rangle + \langle \text{V} \rangle$  ( $\langle \text{Command} \rangle + \langle \text{V} \rangle$  на macOS) — для вставки скопированного текста.

На рис. 1.6 показано, как загрузить скетч Blink в плату Arduino. (Как уже было отмечено ранее, этот скетч загружается в современные платы Arduino при их изготовлении.)

Запустив среду Arduino IDE, выполните последовательность команд меню **Файл | Примеры | 01.Basics** и выберите в списке скетч **Blink** (см. рис. 1.6) — откроется новое окно редактора, содержащее код для мигания встроенным светодиодом (см. рис. 1.5).

Но чтобы плата могла исполнить код скетча, его нужно преобразовать в инструкции, которые может понимать и исполнять главный микроконтроллер платы

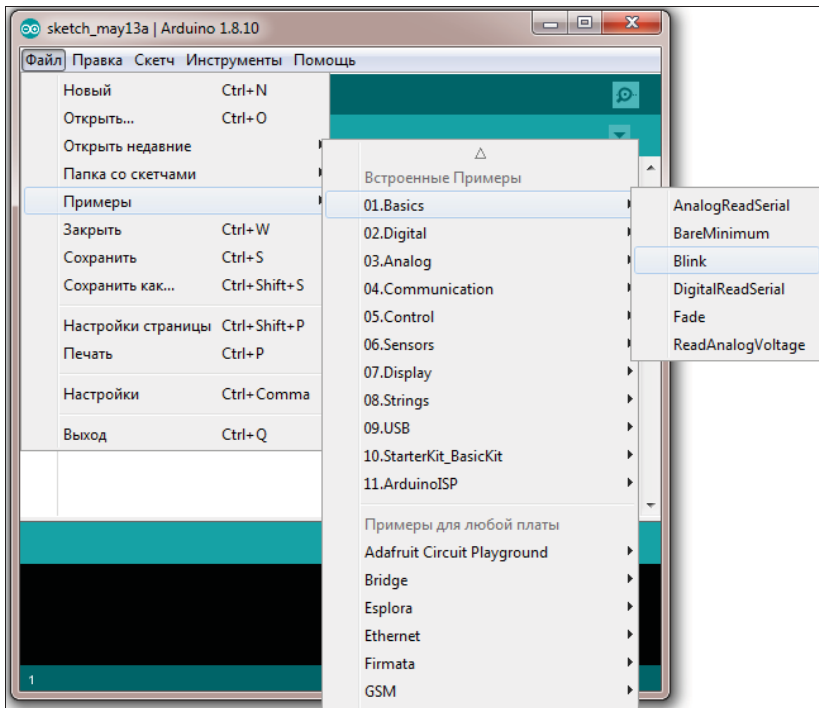


Рис. 1.6. Выбор скетча Blink для загрузки в плату Arduino с помощью меню среды Arduino IDE (на Windows 10)

Arduino. Этот процесс преобразования называется *компиляцией*. Процесс компиляции можно запустить, нажав кнопку **Проверить** (самая левая кнопка окна с галочкой в ней), выполнив последовательность команд меню **Скетч | Проверить/Компилировать** или нажав клавиши быстрого вызова <Ctrl>+<R> (<Command>+<R> на macOS).

В панели сообщений, расположенной ниже окна редактора кода, должно отобразиться сообщение **Компиляция скетча**, а справа от сообщения — индикатор выполнения. Компиляция должна занять пару секунд, после чего в верхней области панели сообщения будет выведено сообщение **Компиляция завершена**, а в самой панели сообщения (область с черным фоном) отобразится более подробное сообщение следующего вида:

Скетч использует 924 байта (2%) памяти устройства. Всего доступно 32 256 байтов. Глобальные переменные используют 9 байтов (0%) динамической памяти, оставляя 2039 байтов для локальных переменных. Максимум: 2048 байтов.

В зависимости от используемой платы Arduino и версии среды разработки Arduino IDE сообщение может быть слегка иным. Но в любом случае в нем предоставляется информация о размере скетча в байтах и максимальном размере скетча, который можно загрузить в плату.

## Обсуждение работы скетча и возможных проблем

Для компиляции скетча среда разработки использует несколько инструментов командной строки. Более подробная информация об этом приводится в *разд. 17.1*.

В сообщении, которое выводится по завершении компилирования скетча, сообщается, какой объем программной памяти платы требуется для хранения скомпилированного кода программы. Если размер скомпилированного скетча превышает объем памяти на плате, будет выведено сообщение следующего вида:

Скетч слишком большой; обратитесь за советами по его уменьшению по адресу:  
<http://www.arduino.cc/en/Guide/Troubleshooting#size>.

В таком случае нужно или уменьшить размер скетча, чтобы он помещался в доступную память на плате, или же использовать другую плату с большим объемом программной флеш-памяти. Возможна и ситуация, когда сам скетч помещается в память, но объем глобальных переменных слишком большой для отведенной для них памяти. При этом выводится сообщение следующего вида:

Недостаточно памяти; обратитесь за советами по адресу:  
<http://www.arduino.cc/en/Guide/Troubleshooting#size>.

Тогда нужно проверить код и уменьшить объем памяти, выделяемый для глобальных переменных, или же использовать плату с большим объемом статической памяти SRAM.



Чтобы не допустить случайного затирания кода примеров, среда Arduino IDE не позволяет сохранять изменения в скетчах примеров, встроенных в среду. Чтобы сохранить измененный скетч примера, ему нужно присвоить другое название, воспользовавшись для этого опцией меню **Файл | Сохранить как**. Собственные скетчи сохраняются с помощью команды меню **Сохранить** или одноименной кнопки (см. *разд. 1.5*).

При наличии в коде синтаксических ошибок компилятор выводит соответствующие сообщения в главной области панели сообщений. Эти сообщения помогают определить причину ошибки, вызвавшей сообщение. Более подробная информация с советами по диагностированию ошибок в скетчах приводится в *приложении 4*.

Кроме сообщений об ошибках, компилятор может выводить предупреждения, если он решит, что некоторые аспекты скетча способны вызвать проблемы. Эти предупреждения могут быть очень полезными, поскольку дают вам возможность принять меры для устранения подозрительных фрагментов кода, которые могут вызвать проблемы в дальнейшем. Пользователь может установить удобный для него уровень предупреждений в настройках, выполнив команду меню **Файл | Настройки** (Window или Linux) или **Arduino | Настройки** (macOS) и выбрав в выпадающем списке **Сообщения компилятора** необходимую опцию: **Ничего**, **По умолчанию**, **Подробнее**, **Все**. Несмотря на наличие в списке опции **По умолчанию**, в действительности по умолчанию устанавливается опция **Ничего**. Рекомендуется установить опцию **По умолчанию** или **Подробнее**.