

Оглавление

Участники	17
Об авторе	17
О рецензентах	17
Предисловие	19
Для кого эта книга	19
Какие темы описаны в книге	19
Как извлечь из книги наибольшую пользу	21
Загрузка файлов с примерами кода	21
Code in Action	22
Изображения в цвете	22
Используемые обозначения	22
Оставайтесь на связи	23
Отзывы	23
ЧАСТЬ 1. ВВЕДЕНИЕ – ОСНОВЫ РОБОТОТЕХНИКИ ...	25
Глава 1. Введение в робототехнику	27
Что такое робот?	27
Продвинутые и впечатляющие роботы	29
Марсоходы	31
Роботы в доме	32
Стиральная машина	32
Другие домашние роботы	33
Роботы в промышленности	34
Роботы-манипуляторы	34
Роботы на складах	36
Роботы для участия в соревнованиях, учебные и любительские роботы	36
Выводы	39
Задание	40
Дополнительные материалы	40
Глава 2. Структурные элементы робота – код и электроника ...	42
Технические условия	42
Внутреннее устройство робота	42
Типы компонентов робота	46
Типы двигателей	46
Другие типы приводов	48

Индикаторы состояния – дисплеи, световые и звуковые индикаторы	49
Типы сенсоров	49
Контроллеры и устройства ввода/вывода	51
Контакты ввода/вывода	52
Контроллеры	53
Модели контроллера Raspberry Pi	56
Набросок компонентов и структуры кода	57
Разработка аппаратного устройства робота	59
Выводы	62
Упражнения	62
Дополнительные материалы	62
Глава 3. Изучение Raspberry Pi	63
Технические требования	63
Функциональные возможности Raspberry Pi	63
Скорость и производительность	64
Возможности подключения	64
Преимущества Raspberry Pi 3A+	64
Выбор способов подключения	65
Контакты для подключения питания	66
Шины передачи данных	67
Входы/выходы общего назначения	67
Платы расширения HAT для Raspberry Pi	67
Операционная система Raspberry Pi	68
Запись образа ОС Raspberry Pi на SD карту	68
Выводы	70
Задание	71
Дополнительные материалы:	71
Глава 4. Автономное управление роботом с помощью Raspberry Pi	72
Технические требования	72
Сущность автономных систем и их преимущества	73
Настройка Wi-Fi и SSH на Raspberry Pi	74
Поиск Raspberry Pi в сети	76
Установка Bonjour для Windows	76
Тест системы	77
Устранение неполадок	77
Подключение к Raspberry Pi посредством PuTTY и SSH	78
Настройка конфигурации ОС Raspberry Pi	80
Изменение имени Raspberry Pi	80
Защита Raspberry Pi	82
Перезагрузка и повторное подключение	82
Обновление программного обеспечения на Raspberry Pi	85
Завершение сеанса Raspberry Pi	86
Выводы	86

Задание.....	87
Дополнительные материалы	87
Глава 5. Создание резервных копий с помощью Git и SD-карты	88
Технические требования.....	88
Причины возникновения проблем при написании и изменении кода	89
Повреждение SD-карты и потеря данных	89
Изменение кода или конфигурации	89
Стратегия 1 – хранение кода на ПК и его последующая передача	90
Стратегия 2 – создание копий с помощью Git.....	92
Стратегия 3 – резервное копирование данных SD-карты	94
Windows.....	95
Mac.....	97
Linux	100
Выводы	101
Задание.....	101
Дополнительные материалы	102
ЧАСТЬ 2. СОЗДАНИЕ АВТОНОМНОГО РОБОТА – ПОДКЛЮЧЕНИЕ ДАТЧИКОВ И ДВИГАТЕЛЕЙ К RASPBERRY PI	103
Глава 6. Сборка основания – колеса, питание и электропроводка	105
Технические требования.....	105
Выбор набора шасси.....	106
Размер	106
Количество колес	107
Колеса и двигатели	108
Простота	109
Стоимость	110
Заключение	110
Выбор платы контроллера двигателя.....	110
Степень интеграции	111
Использование контактов.....	111
Размер	112
Пайка	113
Силовой источник питания	113
Способы подключения	113
Заключение	114
Питание робота.....	115
Проверка компоновки.....	117
Сборка основания.....	120

Установка диска энкодера.....	122
Установка двигателя с помощью опор.....	122
Установка поворотного колеса	124
Установка ведущих колес	125
Прокладка проводов.....	126
Установка Raspberry Pi	127
Установка источников питания.....	128
Готовое основание для робота.....	130
Подключение двигателей к Raspberry Pi.....	131
Подключение проводов к НАТ-плате	132
Автономное питание	134
Выводы	136
Задание.....	137
Дополнительные материалы	137

Глава 7. Движение и повороты – код на Python

для управления двигателями 138

Технические требования.....	138
Разработка кода для проверки двигателей	139
Подготовка библиотек.....	139
Тест – обнаружение НАТ-платы двигателя	139
Тест – демонстрация работы двигателей.....	141
Устранение неполадок	142
Как работает код	142
Рулевое управление робота	144
Типы рулевого управления.....	144
Рулевое управление нашего робота	147
Создание объекта Robot – код для взаимодействия с роботом.....	148
Для чего нужен объект?.....	149
Что мы поместим в объект?.....	150
Разработка сценария для следования по заданной траектории.....	154
Выводы	157
Задание.....	157
Дополнительные материалы	157

Глава 8. Код на Python для работы с датчиками расстояния.....158

Технические требования.....	158
Выбор между оптическими и ультразвуковыми датчиками	159
Оптические датчики.....	160
Ультразвуковые датчики.....	160
Логические уровни и трансляторы логических уровней.....	161
Зачем использовать два датчика?	164
Подключение ультразвуковых датчиков и обработка получаемых ими данных	165
Установка датчиков на шасси	165
Подключение выключателя питания.....	167

Подключение датчиков расстояния	170
Установка библиотек Python для работы с датчиками	172
Считывание данных ультразвуковых датчиков расстояния	172
Устранение неполадок	175
Обход стен – разработка сценария обхода препятствий	176
Добавление датчиков в класс Robot	176
Разработка сценария обхода препятствий	177
Первая попытка разработки стратегии обхода препятствий.....	178
Разработка более сложной стратегии обхода препятствий.....	182
Выводы	184
Задание.....	184
Дополнительные материалы	185
Глава 9. Код на Python для работы со светодиодами	186
Технические требования	186
Что такое линейка RGB-светодиодов?	187
Сравнение технологий светодиодной индикации	187
RGB-значения	189
Подключение светодиодной линейки к Raspberry Pi.....	190
Установка светодиодной линейки на работа	190
Разработка кода для управления дисплеем робота	191
Создание интерфейса для управления светодиодами	191
Добавляем светодиоды в объект Robot	193
Тест одного светодиода	194
Тест всех светодиодов	196
Реализация радужного свечения на светодиодной линейке.....	197
Цветовые модели.....	197
Радужное свечение светодиодов	199
Использование светодиодов для получения отладочных данных в сценариях обхода препятствий	201
Реализация обычного свечения	202
Реализация радужного свечения.....	204
Выводы	205
Задание.....	205
Дополнительные материалы	206
Глава 10. Код на Python для управления сервоприводами	207
Технические требования	207
Что такое сервоприводы?.....	208
Сервопривод: взгляд изнутри	209
Отправка управляющих сигналов сервоприводу.....	210
Позиционирование сервоприводов с помощью Raspberry Pi	212
Создание тестового кода	213
Устранение неполадок	216
Управление ДПТ и сервоприводами	217
Калибровка сервоприводов	218
Создание механизма поворота и наклона.....	219

Сборка механизма	220
Установка механизма поворота и наклона на робота.....	224
Разработка кода для механизма поворота и наклона.....	225
Создание объекта Servos	225
Добавляем сервоприводы в класс Robot	228
Разработка скрипта для движения «головы» робота по окружности	229
Запуск сценария	231
Устранение неполадок	231
Реализация сканирующего сонара	232
Установка датчика	232
Установка библиотек	235
Поведенческий скрипт.....	235
Выводы	238
Задание.....	239
Дополнительные материалы	239
Глава 11. Код на Python для работы с энкодерами	240
Технические требования	240
Измерение пройденного расстояния с помощью энкодеров.....	241
Где применяются энкодеры	241
Типы энкодеров	241
Кодирование абсолютного или относительного положения.....	243
Кодирование направления и частоты вращения	244
Энкодеры, которые мы будем использовать	245
Установка энкодеров на робота.....	246
Подготовка энкодеров.....	247
Снятие Raspberry Pi	248
Установка энкодеров на шасси	249
Подключение энкодеров к Raspberry Pi	249
Измерение пройденного расстояния с помощью кода на Python	251
Ведение журнала	251
Простой подсчет	252
Добавляем энкодеры в объект Robot.....	255
Преобразование тактов в миллиметры	257
Движение по прямой.....	259
Коррекция отклонения с помощью ПИД-регуляторов	259
Создание объекта ПИД-регулятора	261
Создание кода для движения по прямой.....	263
Устранение неполадок	265
Перемещение на определенное расстояние.....	266
Рефакторинг преобразования единиц измерения для класса EncoderCounter.....	266
Настройка констант.....	267
Разработка поведенческого скрипта для движения на определенное расстояние	268
Выполнение точного поворота.....	270
Создание функции drive_arc	274

Выводы	275
Задание.....	276
Дополнительные материалы	276
Глава 12. Код на Python для работы с IMU.....	278
Технические требования	279
Подробнее об инерциальных измерительных модулях (IMU)	279
Предлагаемые модели IMU	280
Пайка – подсоединение контактов к IMU	281
Создание паянного соединения	281
Установка IMU на работа.....	283
Физическое размещение	283
Подключение IMU к Raspberry Pi	286
Считывание данных датчика температуры.....	288
Установка программного обеспечения.....	288
Устранение неполадок	289
Считывание регистра температуры	289
Устранение неполадок	293
Упрощение командной строки VPython	294
Считывание данных гироскопа с помощью Python.....	294
Как работает гироскоп	294
Добавление гироскопа в интерфейс	297
Построение графика данных гироскопа	297
Считывание данных акселерометра с помощью Python	299
Как работает акселерометр.....	299
Добавление акселерометра в интерфейс	300
Отображение данных акселерометра в виде вектора.....	300
Считывание данных магнитометра	302
Как работает магнитометр	302
Добавление интерфейса магнитометра	304
Отображение данных магнитометра	304
Выводы	306
Задание.....	306
Дополнительные материалы	306
ЧАСТЬ 3. СЛУХ И ЗРЕНИЕ – «ИНТЕЛЛЕКТУАЛЬНЫЕ»	
ДАТЧИКИ ДЛЯ РОБОТА.....	309
Глава 13. Система технического зрения робота –	
камера Pi и OpenCV	311
Технические требования	311
Настройка камеры Raspberry Pi.....	312
Установка камеры на механизм поворота и наклона.....	313
Подключение камеры	316
Настройка программного обеспечения для задач	
компьютерного зрения	318

Настройка программного обеспечения камеры Pi	318
Получение изображения с Raspberry Pi	319
Установка OpenCV и вспомогательных библиотек	319
Создание приложения для потоковой передачи данных камеры Raspberry Pi.....	320
Разработка потокового сервера OpenCV.....	321
Создание объекта CameraStream	322
Создание главного приложения сервера изображений.....	323
Создание шаблона	325
Запуск сервера изображений.....	326
Устранение неполадок	326
Запуск фоновых задач во время потоковой передачи.....	327
Создание ядра веб-приложения.....	328
Разработка управляемого поведенческого скрипта	331
Создание шаблона элемента управления	332
Запуск управляемого сервера изображений	333
Отслеживание цветных объектов с помощью кода на Python	333
Преобразование изображения в информацию	334
Усовершенствование ПИД-регулятора	336
Создание компонентов поведенческого скрипта	338
Запуск поведенческого скрипта	343
Настройка параметров ПИД-регулятора	344
Устранение неполадок	345
Отслеживание лиц с помощью кода на Python	346
Поиск объекта на изображении.....	346
Сканирование базовых признаков.....	347
План поведенческого сценария отслеживания лиц.....	349
Создание кода для сценария отслеживания лиц.....	350
Запуск поведенческого скрипта отслеживания лиц	353
Устранение неполадок	354
Выводы	354
Задание.....	355
Дополнительные материалы	355

Глава 14. Код на Python для отслеживания

линий с помощью камеры.....	357
Технические требования.....	358
Введение в отслеживание линий.....	358
Что такое отслеживание линий?.....	358
Использование в промышленности.....	358
Типы алгоритмов отслеживания линий	359
Создание тестового маршрута.....	360
Материалы для создания тестового маршрута	361
Создание линии	361
Конвейерная обработка данных компьютерного зрения для следования по линиям	362

Алгоритмы отслеживания линий с помощью камеры	362
Конвейер обработки изображения	363
Проверка алгоритма компьютерного зрения на основе образцов изображений	366
Зачем нужны тестовые изображения?	366
Получение тестовых изображений	366
Создание кода на Python для обнаружения границ линии	368
Определение положения линии по границам	371
Тест снимков без четкой линии	372
Следование по линиям с помощью ПИД-алгоритма	374
Создание блок-схемы поведенческого сценария	375
Добавляем время в ПИД-регулятор	376
Разработка начального кода для поведенческого скрипта	376
Настройка параметров ПИД-регулятора	383
Устранение неполадок	383
Нахождение линии при ее потере	383
Выводы	384
Задание	385
Дополнительные материалы	385

Глава 15. Голосовая связь с роботом посредством Microsoft..... 386

Технические требования	387
Введение в Microsoft – основные понятия	387
Преобразование речи в текст	387
Слово для пробуждения	387
Реплика	388
Намерение	388
Диалог	388
Словарь	388
Навык	388
Ограничения голосового управления	389
Аудиовход и выход для Raspberry Pi	389
Физическая установка	390
Установка программного обеспечения голосового помощника на Raspberry Pi	391
Установка программного обеспечения ReSpeaker	392
Настройка взаимодействия звуковой карты и Microsoft	394
Начало использования Microsoft	395
Устранение неполадок	397
Разработка API с помощью Flask	397
Как работает управления роботом с помощью Microsoft	398
Удаленный запуск поведенческого скрипта	399
Создание сервера API управления с помощью Flask	401
Устранение неполадок	403
Разработка кода для запуска Microsoft на Raspberry Pi	403
Разработка намерения	404

Устранение неполадок	409
Создание еще одного намерения	410
Выводы	411
Задание.....	412
Дополнительные материалы	413
Глава 16. Погружаемся глубже в работу IMU	414
Технические требования.....	414
Разработка кода для виртуального робота	415
Создание модели робота в VPython	415
Определение параметров вращения с помощью гироскопа.....	419
Калибровка гироскопа	421
Изменение положения виртуального робота на основе данных гироскопа	422
Измерение углов тангажа и крена с помощью акселерометра	426
Определение углов тангажа и крена на основе векторных данных акселерометра	426
Сглаживание данных акселерометра	429
Совместная обработка данных акселерометра и гироскопа	430
Определение направления с помощью магнитометра	433
Калибровка магнитометра	433
Приблизительное определение направления робота с помощью магнитометра на практике	439
Объединение показаний датчиков для ориентирования робота	441
Управление роботом на основе показаний IMU.....	447
Выводы	449
Задание.....	449
Дополнительные материалы	449
Глава 17. Разработка кода на Python для управления роботом с помощью смартфона.....	451
Технические требования.....	452
Когда голосовое управление не работает (или Зачем нам нужно управлять роботом)	452
Режимы меню – выбор поведенческих сценариев	452
Управление режимами робота	454
Устранение неполадок	455
Веб-сервер.....	456
Шаблон	456
Запуск приложения	458
Устранение неполадок	460
Выбор контроллера – как лучше управлять роботом и почему	461
Обзор будущего приложения.....	461
Подготовка Raspberry Pi к удаленному управлению – основы системы управления.....	464
Расширение кода ядра изображения	465
Разработка сценария ручного управления	466

Шаблон (веб-страница)	468
Таблица стилей	471
Разработка кода для ползунков	474
Запуск системы	477
Устранение неполадок	478
Реализация полного управления роботом через смартфон	479
Совместимость режимов меню с поведенческими сценариями, использующими Flask	479
Загрузка видеосервисов	480
Стилизация меню	482
Реализация запуска меню вместе с Pi	484
Добавляем светодиоды на сервер меню	484
Автоматический запуск с помощью инструмента systemd	485
Выводы	487
Задание	488
Дополнительные материалы	488
Часть 4. Дальнейшее изучение робототехники	491
Глава 18. Дальнейшее развитие навыков программирования робототехнических систем	493
Робототехнические интернет-сообщества – форумы и социальные сети	493
Каналы на YouTube, с которыми стоит ознакомиться	495
Технические вопросы – куда обратиться за помощью	496
Мероприятия – соревнования, площадки для совместной работы и встречи	496
Площадки для совместной работы	497
Maker Faire, Raspberry Jam и Coder Dojo	497
Соревнования	498
Навыки, которые стоит освоить, – 3D-печать, пайка, печатные платы и станки с ЧПУ	499
Навыки проектирования	499
Навыки обработки компонентов и сборки	500
Навыки работы с электроникой	502
Подробнее о компьютерном зрении	504
Книги	504
Онлайн-курсы	504
Социальные сети	505
Переход к машинному обучению	505
Операционная система для роботов	506
Выводы	507
Дополнительные материалы	507
Глава 19. Планирование следующего робототехнического проекта – подводим итоги	508
Технические требования	508
Представляем нового робота – как он будет выглядеть	508

Создание блок схемы.....	510
Выбор компонентов	511
Планирование кода	512
Рассказываем миру о проекте	515
Выводы	516
Предметный указатель	517

Участники

ОБ АВТОРЕ

Дэнни Стейпл (Danny Staple) занимается созданием роботов и гаджетов в качестве хобби, снимает видеоролики, посвященные робототехнике, а также является участником таких мероприятий, как Pi Wars и Arduino Day. С 2000 года профессионально занимается разработкой программного обеспечения. В 2009 году начал заниматься программированием на Python, уделяя особое внимание процессам разработки и автоматизации. Большую часть своей профессиональной деятельности Дэнни посвятил работе со встроенными системами, включая встроенные системы Linux. Сейчас является наставником в CoderDojo Nam, где обучает детей программированию. Ранее руководил клубами LEGO Robotics.

Вместе с детьми Дэнни создал таких роботов, как TankBot, SkittleBot, Bangers N Bash (робот из ланчбокса), Big Ole Yellow (еще один гусеничный робот), ArmBot и SpiderBot.

Я хотел бы поблагодарить Дэвида Андерсона (David Anderson) за ценные советы относительно моих идей и мотивацию. Также я выражаю благодарность Бену Нутталлу (Ben Nuttall) и Дейву Джонсу (Dave Jones) (@waveform80) за разработку GPIOZero и ответы на мои бесчисленные вопросы в Twitter. Со встречи с Дейвом Джонсом (Dave Jones), автором библиотеки PiCamera, в ресторане Кардиффа начался мой путь к изучению компьютерного зрения. И наконец, я благодарю своих детей, Хелену (Helena) и Джонатана (Jonathan), за их поддержку, терпение и помощь при работе с графиками.

О РЕЦЕНЗЕНТАХ

Лео Уайт (Leo White) – выпускник Кентского университета (University of Kent), профессиональный инженер-программист, интересующийся электроникой, 3D-печатью и робототехникой. Сначала он занимался программированием на Commodore 64, затем разработал несколько приложений для Acorn Archimedes, а сейчас на постоянной основе занимается разработкой кода для ТВ-приставок. На основе Raspberry Pi Лео создавал роботов из детских игрушек и роботов-манипуляторов, попутно описывая этот процесс и свой опыт в блоге. Также проводил презентации на Raspberry Jams и участвовал в соревнованиях Pi Wars.

Рамкумар Гандинатан (Ramkumar Gandhinathan) – исследователь и профессиональный робототехник. Своего первого робота создал в шестом классе. Благодаря личным и профессиональным связям он занимается робототехникой уже более 15 лет. Рамкумар создал более 80 различных роботов. Общий

профессиональный опыт в области робототехники составляет 7 лет (4 года полной занятости и 3 года неполной/стажировки). На протяжении 5 лет он работал с ROS (Robot Operating System – операционная система для роботов). В рамках профессиональной карьеры Рамкумар разработал свыше 15 решений ROS для промышленных роботов. Также он увлекается созданием и пилотированием дронов. В круг его исследовательских интересов и увлечений входит SLAM (Simultaneous Localization And Mapping – одновременная локализация и построение карты), планирование движений, совместное использование данных датчиков, коммуникация между роботами и системная интеграция.

Предисловие

Эта книга посвящена созданию интеллектуального робота и разработке кода для его поведенческих сценариев. Вы узнаете о навыках, которые требуются для создания робота из отдельных компонентов, а также о тонкостях их выбора. Для построения робота из этой книги вы будете использовать такие компоненты, как датчики, двигатели, камеры, микрофоны, динамики, светодиоды и Raspberry Pi.

Далее в книге вы узнаете, как разработать код для перечисленных компонентов. Вы будете использовать язык программирования Python, а также немного HTML/CSS и JavaScript.

Все компоненты робота, описанные в этой книге, широко доступны. Приведенные примеры кода призваны продемонстрировать, как работает та или иная технология. В дальнейшем вы сможете комбинировать разные части программного и аппаратного обеспечения, чтобы создавать сложных роботов с более интересными поведенческими сценариями.

В книге пересекаются темы программирования и робототехники, а также раскрывается ряд специализированных тем, таких как компьютерное зрение и голосовое управление.

Для кого эта книга

Книга подойдет как новичкам в области программирования, так и опытным программистам, желающим применить свои навыки в аппаратном проекте. Для этого не обязательно быть экспертом, достаточно быть способным ввести несколько строк кода и уметь работать с циклами, условиями и функциями. В книге затрагивается тема объектно-ориентированного программирования, но вам не обязательно разбираться в нем перед прочтением.

Для создания робота из этой книги вам не понадобится специально оборудованная мастерская – достаточно уметь паять и крепить компоненты болтами. Позже мы поговорим об этом подробнее.

Вам вовсе не обязательно иметь опыт работы с электроникой или созданием каких-либо устройств. В книге представлены базовые понятия, и я надеюсь, что это вызовет у вас здоровый интерес к дальнейшему изучению. Самое главное – это ваше желание создать робота и научить его делать интересные вещи.

Какие темы описаны в книге

В главе 1 рассказывается о том, как устроены роботы, как они применяются на производстве и в быту, а также описываются некоторые примеры роботов, созданных новичками.

В главе 2 подробно рассказывается о компонентах робота и о том, как их выбирать. Также здесь представлены блок-схемы будущей системы и кода.

В главе 3 говорится о микрокомпьютере Raspberry Pi, его подключении и операционной системе – Raspbian Linux. В этой главе вы запишете образ операционной системы на SD-карту и научитесь использовать ее в работе.

В главе 4 рассказывается, как реализовать беспроводную связь робота с Raspberry Pi.

В главе 5 вы узнаете, какие неполадки могут возникнуть в коде и как предотвратить его потерю, сохранив копии в репозиториях.

В главе 6 мы переходим к сборке основания робота. Здесь рассказывается о тонкостях выбора компонентов и проверке их размещения.

В главе 7 показано, как разработать код для движения робота. Также здесь закладываются основы для кода из следующих глав.

В главе 8 мы добавим роботу датчики и разработаем для них код, благодаря которому робот сможет самостоятельно обходить стены и препятствия.

В главе 9 рассказывается, как добавить роботу цветные светодиоды. Здесь вы узнаете, как использовать дополнительные выводы для отладки или просто для развлечения.

В главе 10 рассказывается о сервоприводах и их использовании для позиционирования сенсорной головки, а также «конечностей» робота.

В главе 11 показано, как создать код для получения точной информации о перемещении колес с помощью одометра/тахометра. Здесь ваш робот научится ездить по прямой, выполнять точные повороты и определять, насколько далеко он переместился. В этой главе мы впервые поговорим о ПИД-регуляторах.

В главе 12 рассказывается об **инерциальном измерительном модуле (IMU)**, который включает датчики для измерения температуры, ускорения, скорости вращения и магнитных полей. В этой главе мы впервые поговорим о пайке и VPython.

В главе 13 вы узнаете, как получать данные с камеры, и научите робота передвигаться, основываясь на компьютерном зрении. Также здесь показано, как осуществляется потоковая передача обработанного видео в браузер.

В главе 14 рассказывается, как научить робота следовать по линиям с помощью камеры Raspberry Pi.

В главе 15 описывается разработка системы управления роботом с помощью голосового помощника. Вы научитесь использовать голосовые команды и получать от робота ответы.

В главе 16 рассказывается об объединении датчиков, описанных в главе 12. Вы узнаете, как использовать их для определения положения робота, а также разработаете поведенческий скрипт, который будет действовать как компас.

В главе 17 мы переходим к созданию веб-приложения меню и панели управления, похожей на геймпад. С помощью этой системы вы сможете управлять роботом со смартфона, опираясь на видео с камеры.

В главе 18 вы узнаете больше о мире робототехники. Здесь рассказывается о различных робототехнических сообществах, потенциальных областях развития и соревнованиях роботов.

В заключительной главе 19 обобщается представленная в книге информация, а также предлагаются варианты по созданию следующего робототехнического проекта.

КАК ИЗВЛЕЧЬ ИЗ КНИГИ НАИБОЛЬШУЮ ПОЛЬЗУ

Прежде чем начать работать с книгой, я рекомендую вам потренироваться в работе с текстовыми языками программирования. Познакомьтесь с основными переменными, условными операторами, циклами и функциями.

Для работы вам потребуется компьютер с операционной системой macOS, Linux или Windows, доступ к интернету и Wi-Fi.

Что касается навыков ручного труда, я думаю, что вы умеете пользоваться отверткой и без проблем справитесь с кропотливой работой. Также я надеюсь, что вас не пугает пайка.

Представленные в книге примеры кода были протестированы на Python 3 с помощью Raspbian Buster и Picroft Buster Keaton. Их установка описывается в тексте соответствующих разделов. Также в тексте рассказывается, как выбрать нужные аппаратные компоненты.

Программное/аппаратное обеспечение, используемое в книге	Требования к операционной системе
Python 3	Raspbian Buster
Picroft/Mycroft	Picroft Buster Keaton
OpenCV	Raspbian Buster/Python 3
VPython	Raspbian Buster/Python 3
Flask	Python 3

Прежде чем приобрести какие-либо аппаратные компоненты, ознакомьтесь с главами, в которых рассказывается, как правильно их выбрать.

Если вы используете электронную версию книги, я рекомендую вводить код самостоятельно или брать его из репозитория на GitHub (ссылка предоставлена в следующем разделе). Это позволит избежать ошибок, связанных с неправильным копированием/вставкой фрагментов кода.

ЗАГРУЗКА ФАЙЛОВ С ПРИМЕРАМИ КОДА

Все файлы с кодом из этой книги доступны через ваш аккаунт на веб-сайте www.packt.com. Если вы приобрели книгу в другом месте, зарегистрируйтесь, перейдите по адресу <https://www.packtpub.com/support> и оставьте запрос на отправку файлов по электронной почте.

Загрузить файлы с кодом вы можете следующим образом.

1. Войдите (или зарегистрируйтесь) в аккаунт на веб-сайте www.packt.com.
2. Перейдите во вкладку **Support** (Поддержка).
3. Нажмите **Code Downloads** (Загрузить код).
4. В строке поиска введите название книги и следуйте дальнейшим инструкциям на экране.

После загрузки файлов нужно обязательно распаковать или извлечь папки с помощью новейшей версии одной из следующих программ:

- WinRAR/7-Zip для Windows;
- Zipeg/iZip/UnRarX для Mac;
- 7-Zip/PeaZip для Linux.

Комплект исходного кода для этой книги размещен на GitHub по адресу <https://github.com/PacktPublishing/Learn-Robotics-Programming-Second-Edition>. В случае обновления кода репозиторий также обновится.

По адресу <https://github.com/PacktPublishing/> доступны другие комплекты исходного кода из нашего богатого каталога книг и видео. Ознакомьтесь и с ними!

CODE IN ACTION

Видеоролики Code in Action для этой книги доступны по адресу <http://bit.ly/3bu5wHp>.

ИЗОБРАЖЕНИЯ В ЦВЕТЕ

По адресу https://static.packt-cdn.com/downloads/9781839218804_ColorImages.pdf вы можете скачать файл, содержащий все изображения (в том числе скриншоты и схемы) из книги, в формате PDF.

ИСПОЛЬЗУЕМЫЕ ОБОЗНАЧЕНИЯ

В книге используется ряд текстовых обозначений.

Код в тексте: так в тексте выделяются фрагменты кода, названия таблиц баз данных, папок, файлов, расширения файлов, составные имена файлов, пути, фиктивные URL-адреса, фрагменты пользовательского ввода и ссылки на Twitter. Например: «... задает параметр `color` для светодиода с номером `led_number`».

Блоки кода выглядят следующим образом:

```
cyan_rgb = [int(c * 255) for c in cyan]
```

Элементы или строки кода, на которые нужно обратить особое внимание, выделяются жирным шрифтом:

```
right_distance = self.robot.right_distance_sensor.distance
# Отображение
self.display_state(left_distance, right_distance)
```

Ввод или вывод командной строки выглядит так:

```
>>> r.leds.show()
```

Жирный шрифт: так выделяются новые термины, важные части текста или текст, который вы должны видеть на экране. Например, текст из меню или диалоговых окон: «Выберите **4** для **Other USB Microphone** (Другой USB-микрофон) и проверьте звук».

Советы и важные примечания

Они выделяются следующим образом.

ОСТАВАЙТЕСЬ НА СВЯЗИ

Мы всегда рады обратной связи.

Общие вопросы: любые вопросы насчет этой книги вы можете задать в электронном письме по адресу customercare@packtpub.com, указав в теме письма название книги.

Ошибки и опечатки: мы приложили все усилия, чтобы содержание было точным, однако ошибки все же случаются. Если вы нашли ошибку в этой книге, сообщите нам об этом. Для этого перейдите по адресу www.packtpub.com/support/errata, выберите нужную книгу, щелкните по ссылке **Errata Submission Form** (Форма отправки сведений об ошибках) и введите ваше сообщение.

Нарушение авторского права: если вы столкнетесь с незаконными копиями наших книг, размещенными в интернете в любой форме, сообщите нам адрес или название веб-сайта. Для этого отправьте письмо на электронный адрес copyright@packt.com со ссылкой на материал.

Для авторов: по вопросам сотрудничества авторы могут обратиться по адресу <https://authors.packtpub.com>.

ОТЗЫВЫ

Почему бы не оставить отзыв о прочитанной книге на сайте, где вы ее приобрели? Благодаря вам потенциальные читатели перед приобретением смогут ознакомиться с вашим непредвзятым мнением, а издательство Packt и авторы увидят отзыв о своем продукте. Спасибо!

Больше информации на веб-сайте www.packt.com.



Часть 1

Введение – основы робототехники

В этой части вы на примерах рассмотрите, что представляет из себя робот, узнаете, как он устроен, и увидите, как подготовить Raspberry Pi для дальнейших экспериментов.

Часть включает в себя следующие главы:

- главу 1. Введение в робототехнику;
- главу 2. Структурные элементы робота – код и электроника;
- главу 3. Изучение Raspberry Pi;
- главу 4. Автономное управление роботом с помощью Raspberry Pi;
- главу 5. Создание резервных копий с помощью Git и SD-карты..

Глава 1

Введение в робототехнику

В этой книге мы расскажем, как построить робота и создать программы, позволяющие ему имитировать разум и способность принимать самостоятельные решения. Мы напишем код для сенсоров, благодаря которому робот сможет наблюдать за окружающей средой. Также посредством кода мы наделим робота такими способностями, как зрение, речь и распознавание речи.

Вы увидите, как путем комбинации простых методов сборки и небольшого количества кода можно создать робота, по многим признакам напоминающего настоящее домашнее животное. Кроме того, мы расскажем об отладке роботов подобного типа, что пригодится вам в случае, если что-то пойдет не так (к слову, это неизбежно). Вы узнаете, как обучить робота давать сигналы об ошибках, а также выбирать поведение, соответствующее вашим установкам. Мы подключим к роботу джойстик с функцией голосового управления и, наконец, покажем как организовать дальнейшую сборку робота.

Прежде чем приступать к созданию робота, необходимо разобраться в том, что он из себя представляет. Здесь мы рассмотрим некоторые типы роботов, а также выделим особенности, отличающие их от других машин. Вы попробуете провести грань между роботами и другими машинами. Однако, когда вы узнаете, как обстоит ситуация на самом деле, все может стать чуть более запутанным. Далее мы рассмотрим ряд уже существующих роботов, созданных робототехниками-любителями.

В этой главе будут разобраны следующие темы:

- что такое робот;
- продвинутые и впечатляющие роботы;
- роботы в доме;
- роботы в промышленности;
- роботы для участия в соревнованиях, учебные и любительские роботы.

Что такое робот?

Робот – это машина, способная принимать автономные решения на основе данных сенсоров. Программный агент – это программа, обеспечивающая автоматическую обработку входных данных и выполнение выходных операций. Возможно, лучше всего можно описать робота как автономного программного агента с сенсорами, способного передвигаться в соответствии с выходными операциями. Часто роботов характеризуют как электромеханические платформы с программным обеспечением. Так или иначе, робот состоит из трех основных элементов: электроники, механических деталей и кода.

Слово «*робот*» вызывает в воображении образы творений из мира фантастики и научной фантастики, машин, обладающих невероятной силой и интеллектом. Зачастую эти творения очень похожи на людей, т. е. являются **андроидами**. Авторы наделяют их индивидуальностью, ввиду чего роботы ведут себя как несколько наивные люди:

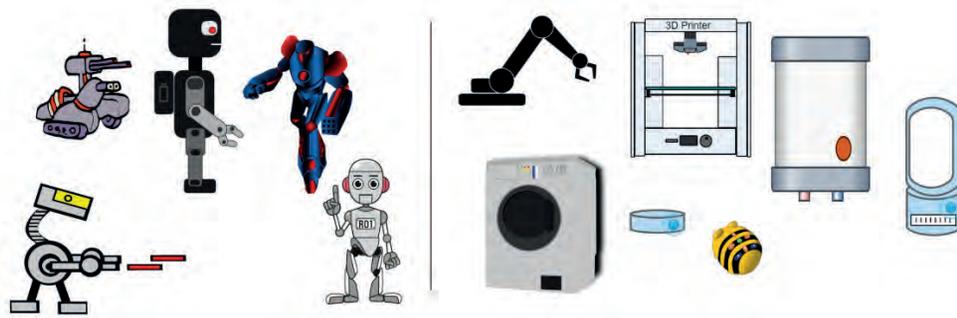


Рис. 1.1. Роботы из мира научной фантастики и реальные роботы. Используемые изображения взяты из публичной библиотеки OpenClipArt

Название «*робот*» пришло из научно-фантастических произведений. Оно происходит от чешского слова, означающего «*раб*». Впервые название было использовано в пьесе Карела Чапека (Karel Čapek) «*Россумские универсальные роботы*» (*Rossum's Universal Robots*), поставленной в 1921 году. Слово «*робототехника*» придумал автор научно-фантастических произведений Айзек Азимов (Isaac Asimov) во время исследования поведения «интеллектуальных» роботов.

Большинство роботов, используемых человеком, на самом деле не являются столь продвинутыми и притягивающими внимание. Большинство из них не стоит на двух ногах, да и вообще не имеет ног. Некоторые из них передвигаются с помощью колес, а некоторые не передвигаются и вовсе, но при этом имеют движущиеся части и сенсоры.

Такие роботы, как стиральные машины, роботы-пылесосы, полностью саморегулирующиеся бойлеры и вентиляторы с функцией забора проб воздуха, проникли в наши дома и стали частью повседневной жизни. Они не несут угрозы и являются просто машинами вокруг нас. Гораздо больший интерес вызывают 3D-принтеры, роботы-манипуляторы и обучающие игрушки:

По своей сути любой робот может быть представлен в виде трех составляющих: элемента, отвечающего за выходные операции (**выход**), например двигателя; элемента, осуществляющего сбор входных данных (**вход**), например сенсора; и **контроллера**, обеспечивающего обработку данных и исполнение кода. Итак, базовый робот выглядит следующим образом.

- Он оснащен входами и сенсорами, предназначенными для измерения свойств окружающей среды.
- Он имеет выходы, например двигатели, световые и звуковые индикаторы, клапаны и нагревательные элементы, способные влиять на окружающую среду.
- Он принимает решения относительно выходных операций, основываясь на полученных входных данных.

В следующем разделе мы рассмотрим более продвинутых роботов.

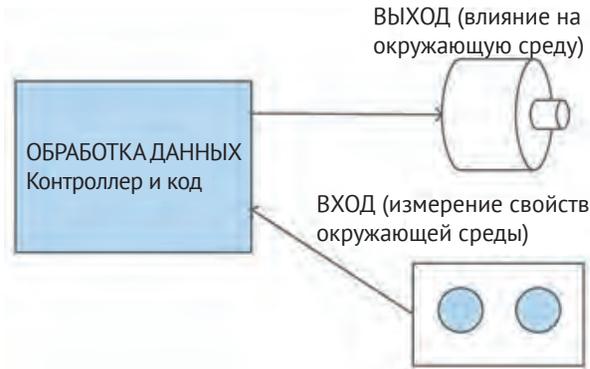


Рис. 1.2. Упрощенное представление робота

ПРОДВИНУТЫЕ И ВПЕЧАТЛЯЮЩИЕ РОБОТЫ

Теперь, когда вы имеете общее представление о роботах, я представлю несколько наиболее впечатляющих примеров роботов и расскажу об их способностях. Помимо марсоходов, зачастую создатели роботов отдают предпочтение роботам, похожим на людей и животных. Такой выбор обусловлен большей адаптивностью подобных форм, в отличие от роботов, разработанных для промышленного использования и предназначенных для многократного выполнения однообразных задач (*single repeated use*).

На рис. 1.3 представлены роботы, похожие на людей или животных.

Общим для представленных роботов является то, что они имитируют людей и животных.

1. Робот 1 – это Cog, созданный командой Массачусетского технологического института (Massachusetts Institute of Technology, MIT). Cog должен был походить на человека своими движениями и «органами восприятия» (сенсорами).
2. Робот 2 – это робот-андроид ASIMO, разработанный корпорацией Honda. Он умеет двигаться и разговаривать почти как человек. ASIMO оснащен двумя камерами, служащими для *обхода препятствий*, распознавания жестов и лиц. Также робот способен определять поверхность посредством лазерных датчиков расстояния. Благодаря инфракрасным сенсорам он может обнаруживать отметки на полу и следовать по ним. ASIMO понимает голосовые команды на английском и японском языках.
3. Робот 3 – Nao, созданный Softbank Robotics. Этот симпатичный робот высотой всего 58 см был разработан как обучающее и игровое программируемое устройство. Он оснащен сенсорами, посредством которых может следить за собственными движениями (например, распознавать падение) и ультразвуковыми датчиками, позволяющими *предотвращать столкновения* с другими объектами. Nao может распознавать и обрабатывать голосовые команды с помощью динамиков и микрофона. Он, как и ASIMO, оснащен несколькими камерами для схожих целей.

4. Робот 4 – Atlas от Boston Dynamics. Этот робот отличается скоростью перемещения на двух ногах и плавностью движений. Он оснащен матрицей лазерных сканеров – **лидаров** (LIDAR), благодаря которым может обнаруживать окружающие его объекты, что позволяет планировать дальнейшее движение и предотвращать столкновения.
5. Робот 5 – это четвероногий робот BigDog, также созданный Boston Dynamics. От других подобных роботов он отличается устойчивостью, способностью оставаться в вертикальном положении при толчках и не падать при движении по льду.

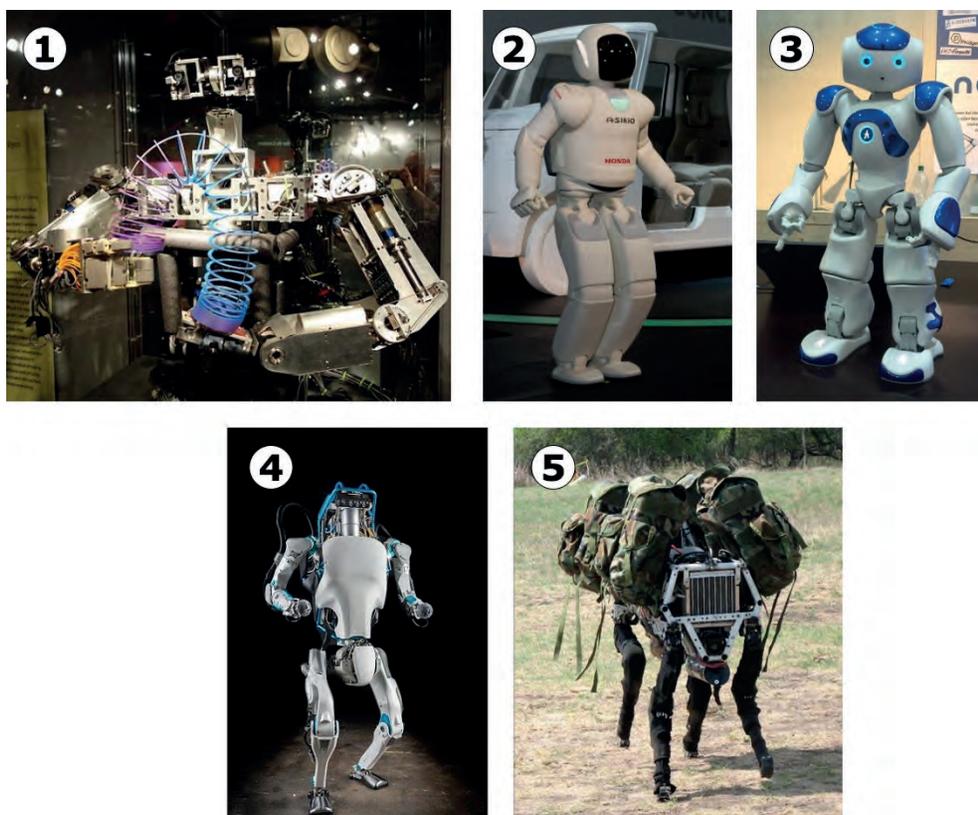


Рис. 1.3. Подборка роботов, похожих на людей и животных. [Источники графических материалов: Изображение 1: https://commons.wikimedia.org/wiki/File:Cog,_1993-2004,_view_2_-_MIT_Museum_-_DSC03737.JPG (открытый доступ); Изображение 2: [https://commons.wikimedia.org/wiki/File:Honda_ASIMO_\(ver._2011\)_2011_Tokyo_Motor_Show.jpg](https://commons.wikimedia.org/wiki/File:Honda_ASIMO_(ver._2011)_2011_Tokyo_Motor_Show.jpg), автор Morio, CC BY-SA 3.0: <https://creativecommons.org/licenses/by-sa/3.0/deed.en>; Изображение 3: это [https://commons.wikimedia.org/wiki/File:Nao_Robot_\(Robocup_2016\).jpg](https://commons.wikimedia.org/wiki/File:Nao_Robot_(Robocup_2016).jpg) (открытый доступ); Изображение 4: [wikimedia.org/wiki/File:Atlas_from_boston_dynamics.jpg](https://www.kansascity.com/news/business/technology/917xpi/picture62197987/ALTERNATES/FREE_640/atlas%20from%20boston%20dynamics), автор https://www.kansascity.com/news/business/technology/917xpi/picture62197987/ALTERNATES/FREE_640/atlas%20from%20boston%20dynamics, CC BY-SA 4.0: <https://creativecommons.org/licenses/by-sa/4.0/deed.en>; Изображение 5: https://commons.wikimedia.org/wiki/Commons:Licensing#Material_in_the_public_domain (открытый доступ)

Вы сможете реализовать подобные функции в своем роботе. Мы обсудим применение датчиков расстояния с целью обхода препятствий, рассмотрим ультразвуковые датчики расстояния, как у Nao, а также лазерные датчики расстояния, как у ASIMO. Далее мы исследуем применение камер для обработки видеоданных, линейных сенсоров для отслеживания отметок на полу, а также обработку голоса для реализации управления голосовыми командами. Также мы разработаем механизм поворота и наклона камеры, как у Cog.

Марсоходы

Роботы-марсоходы предназначены для работы на другой планете, где в случае поломки не сможет вмешаться человек. Они обладают прочной конструкцией. Обновление программного обеспечения марсоходов осуществляется удаленно, поскольку нецелесообразно отправлять с этой целью на Марс человека с экраном и клавиатурой. По своей конструкции марсоходы предназначены для полностью автономной деятельности :



Рис. 1.4. Марсоход НАСА Curiosity в Глен-Этив, Марс (Источник: NASA/JPL-Caltech/MSSS; <https://mars.nasa.gov/resources/24670/curiosity-at-glen-etive/?site=msl>)

Марсоходы передвигаются с помощью колес, а не ног, поскольку стабилизировать колесного робота намного проще. Также в связи с этим снижаются риски. Каждое колесо марсохода имеет собственный двигатель. Колеса расположены таким образом, чтобы обеспечить максимальное сцепление и устойчивость при движении по каменистой поверхности Марса в условиях пониженной гравитации.

Марсоход Curiosity оснащен высокочувствительной камерой, которая в момент отправки была сложена. После приземления на Марс камера была развернута посредством сервоприводов. Камера наводится с помощью **механиз-**

ма поворота и наклона. Главная задача здесь – охватить как можно больше фрагментов ландшафта Марса и отправить отснятые материалы и фотографии в НАСА (National Aeronautics and Space Administration – NASA) для анализа.

Подобно марсоходам, робот, которого вы создадите в результате прочтения этой книги, будет передвигаться с помощью колес с приводом от двигателя. Наш робот предназначен для работы без устройств ввода, поскольку его конструкция не предусматривает прямое участие человека в его работе. По мере расширения возможностей нашего робота мы также будем использовать сервоприводы для управления механизмом поворота и наклона.

РОБОТЫ В ДОМЕ

Многие роботы уже проникли в наши дома. Мы не воспринимаем их как таковых, поскольку на первый взгляд они кажутся довольно обыденными и приземленными. Однако на самом деле они сложнее, чем кажется.

Стиральная машина

Для начала поговорим о стиральной машине, которая сегодня есть почти в каждом доме. Каждый день мы используем этот бытовой прибор для стирки, отжима и сушки одежды. Но можно ли считать его роботом?

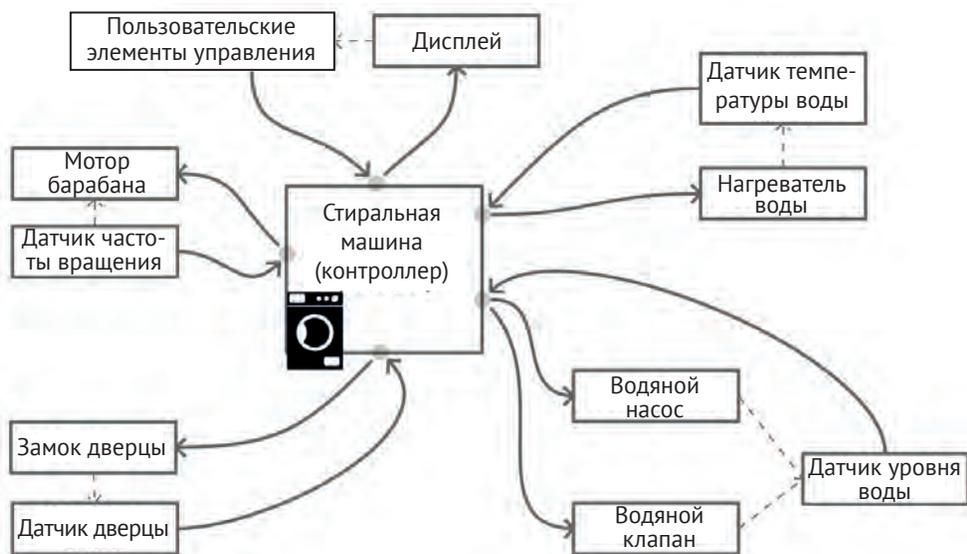


Рис. 1.5. Устройство стиральной машины

На рис. 1.5 схематично представлено устройство стиральной машины. Здесь мы видим центральный контроллер, который подключается к дисплею с элементами управления для выбора программы. Линии, выходящие из контроллера, являются выходами, а входящие представляют собой данные, поступающие с сенсоров (входы). Пунктирными линиями, идущими от выходов к входам, обозначен замкнутый цикл выходных операций, происходящих в ре-

альном мире и воздействующих на сенсоры. Данный цикл называется *петлей обратной связи*, или просто *обратной связью*, и является одной из важнейших концепций в робототехнике.

Пользователь выбирает настройки стиральной машины путем нажатия кнопок, а статус процесса отображается на дисплее. После нажатия кнопки запуска контроллер проверяет датчик дверцы, и в случае, если дверца не закрыта, процесс стирки не запустится. При запуске стирки дверца блокируется. Затем посредством нагревателей, клапанов и насосов, барабан стиральной машины заполняется водой. Путем обратной связи по сигналам датчика регулируется уровень и температура воды.

```

запустить водяной насос
включить нагреватель воды
while барабан not заполнен and вода not достаточно горячая:
    if барабан заполнен
        остановить водяной насос
    if вода достаточно горячая
        выключить нагреватель воды
else
    включить нагреватель воды

```

Каждый процесс можно представить в виде набора строк, представленного выше. Подобный набор строк отвечает за одновременное наполнение барабана и поддержание температуры воды.

Обратите внимание, что строка после `else` необходима для тех случаев, когда температура воды падает ниже установленной. Далее стиральная машина начинает вращать барабан – сначала медленно, а затем быстро, при этом определяя оптимальную скорость для выбранной программы. После стирки вода из барабана сливается и стиральная машина отжимает одежду. Затем снимается блокировка дверцы, и процесс завершается.

Стиральную машину можно назвать роботом во всех отношениях. Это устройство имеет сенсоры, на основе данных которых оно способно выполнять выходные операции. Посредством обработки сигналов, полученных от сенсоров с обратной связью, контроллер следует установленной пользователем программе. Так, любой мастер по ремонту стиральных машин может быть даже ближе к робототехнике, чем я.

Другие домашние роботы

Газовый бойлер, используемый вместо центрального отопления (gas central heating boiler), оснащен сенсорами, насосами и клапанами. Посредством механизмов обратной связи такой бойлер может поддерживать температуру в доме, регулировать расход воды на отопление, а также внимательно следить за тем, чтобы горелка не погасла. Бойлеры во многом похожи на роботов. Однако они являются стационарными, ввиду чего не могут быть адаптированы для иных целей. То же самое можно сказать и о других бытовых приборах, таких как умные вентиляторы и принтеры.

Умные вентиляторы определяют температуру, влажность и качество воздуха в помещении посредством сенсоров. На основании полученных данных они регулируют скорость вращения и нагревательные элементы.

Многие бытовые приборы, например такие как микроволновая печь, умеют работать только по таймеру. Такие устройства крайне просты и не способны принимать самостоятельные решения. Следовательно, их нельзя отнести к роботам.

На рис. 1.6 представлен, пожалуй, наиболее знакомый нам домашний робот – робот-пылесос.



Рис. 1.6. Робот-пылесос PicaBot (Источник: Handitec [Открытый доступ – [https:// commons.wikimedia.org/wiki/File:PicaBot.jpg](https://commons.wikimedia.org/wiki/File:PicaBot.jpg)])

Этот колесный мобильный робот похож на тот, которого мы предлагаем построить в этой книге, но чуть более симпатичный. Робот-пылесос оснащен сенсорами для обнаружения стен, ступеней и ограниченных зон. Также сенсоры помогают ему избегать столкновений. Как и наш будущий робот, робот-пылесос обладает такими характеристиками, как автономность, мобильность и программируемость.

По мере создания нашего робота мы научимся использовать сенсоры для обнаружения объектов. Также наш робот сможет реагировать на препятствия посредством формирования петель обратной связи, подобных тем, которые мы рассматривали в подразделе о стиральных машинах.

РОБОТЫ В ПРОМЫШЛЕННОСТИ

Еще одно место, где часто встречаются роботы, – это промышленность. Первые роботы появились на заводах уже очень давно.

Роботы-манипуляторы

Роботы-манипуляторы применяются для самых разных целей. Это могут быть небольшие роботы для переворачивания яиц или исполинские устрой-

ства, способные перемещать транспортировочные контейнеры. Как правило, роботы-манипуляторы оснащены шаговыми двигателями и сервоприводами. В этой книге мы рассмотрим серводвигатели в механизмах поворота и наклона. Большинство промышленных роботов-манипуляторов (например, сварочные роботы от компании ABB) следуют заранее заданной схеме движений и не принимают самостоятельные решения. Однако среди промышленных роботов встречаются и другие, представляющие собой интеллектуальные системы, основанные на сенсорах. Например, робот-манипулятор **Baxter** от Rethink Robotics, представленный на рис. 1.7.



Рис. 1.7. Робот-манипулятор Baxter от Rethink Robotics (Источник: Baxter на выставке Innorobo, © Ксавье Капе (Xavier Caré)/Wikimedia Commons [CC-BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>)])

Работать рядом с многими роботами-манипуляторами небезопасно, поскольку это может привести к несчастным случаям. Поэтому подобные роботы зачастую размещаются за ограждениями, а рядом с ними наносится предупредительная маркировка. Однако Baxter несколько отличается от них: он умеет распознавать нахождение человека поблизости, а также может останавливаться во избежание несчастного случая. На рис. 1.7 вы можете рассмотреть сенсоры, отвечающие за упомянутые функции. Они расположены вокруг «головы» робота. Сенсоры, расположенные на «руках», а также нежесткие сочленения позволяют роботу обнаруживать столкновения и реагировать на них.

Baxter оснащен механизмом обучения и повторения, благодаря которому сотрудники производства могут адаптировать его к различным задачам. При тренировке или воспроизведении движений Baxter определяет и фиксирует нужные позиции. Наш робот будет использовать *кодовые датчики* (их часто называют просто *энкодерами*) для управления движением колес.

Роботы на складах

Другой распространенный тип роботов в промышленности – это роботы, служащие для транспортировки предметов по цеху или складу.



Рис. 1.8. Роботизированные системы на складах: система Stingray от TGWmechanics [CC BY-SA 3.0 (<https://creativecommons.org/licenses/by-sa/3.0/>)] и система Intellicart от Mukeshhrs [открытый доступ]

Под номером 1 на рис. 1.8 представлены роботизированные системы подъемных кранов, предназначенные для транспортировки поддонов в складских комплексах. Они осуществляют транспортировку внутри стеллажных систем согласно получаемым инструкциям.

Под номером 2 на рис. 1.8 представлены небольшие роботы Intellicart, также предназначенные для перемещения предметов. Они полагаются на линейные сенсоры, ориентируясь на линии на полу, магнитно-чувствительные провода под полом или маркерные маяки (как ASIMO). Наш робот также сможет следовать по линиям. Как правило, все подобные роботы являются колесными, поскольку такой вариант наиболее прост в обслуживании, а также позволяет создавать стабилизированные платформы.

РОБОТЫ ДЛЯ УЧАСТИЯ В СОРЕВНОВАНИЯХ, УЧЕБНЫЕ И ЛЮБИТЕЛЬСКИЕ РОБОТЫ

Самые занятые и инновационные роботы обычно создаются робототехниками-любителями.

Робототехника заняла прочную позицию в образовании. Многие робототехники, занятые в сфере образования, применяли свои разработки для обучения и экспериментов. Из таких учебных проектов впоследствии сформировалось множество коммерческих проектов. Университетские роботы, как правило, являются продуктом коллективного труда. Студентам и научным сотрудникам, занимающимся разработкой роботов, открывается доступ к высокотехнологичному оборудованию:

Kismet (под номером 1 на рис. 1.9) был создан командой Массачусетского технологического института (Massachusetts Institute of Technology – MIT) в конце 90-х годов. На его основе впоследствии было создано несколько любими-

тельских роботов. Для своего времени Kismet был новаторской разработкой. Посредством приводов он мог управлять движениями и имитировать человеческую мимику. В частности, на основе Kismet создан OhBot (под номером 2 на рис. 1.9) – бюджетный любительский проект. Этот робот также может имитировать человеческую мимику и делает это под управлением Raspberry Pi на основе распознавания голоса и обработки данных камеры.

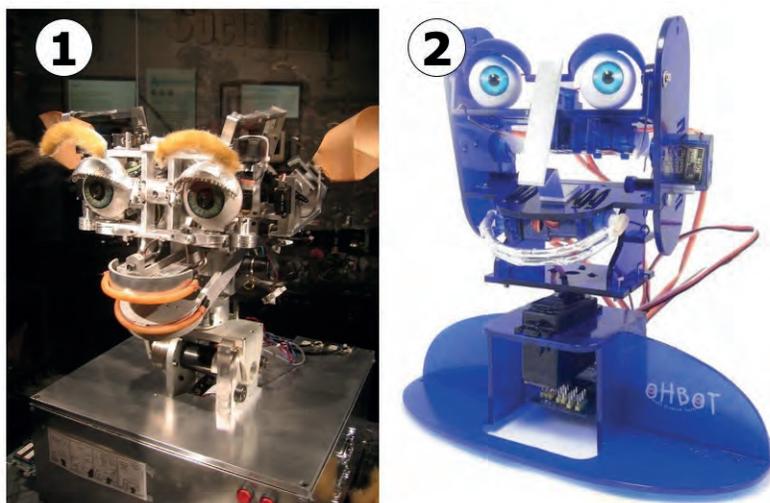


Рис. 1.9. Kismet [Джаред С. Бенедикт (Jared C Benedict) CC BY-SA 2.5 <https://creativecommons.org/licenses/by-sa/2.5>] и OhBot [AndroidFountain [CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0>)]]

Любительская робототехника тесно связана с сообществом открытого программного и аппаратного обеспечения. На таких веб-сайтах, как GitHub (<https://github.com>), любители делятся своими проектами и кодами, что рождает новые идеи. Любительские роботы могут создаваться на основе готовых проектов в интернете, но с некоторыми модификациями и дополнениями. На сегодняшний день в интернете доступно множество наборов для сборки роботов, начиная от простых трехколесных платформ до дронов и гексаподов¹. В некоторые наборы входят как механические, так и электронные компоненты. Подробнее мы рассмотрим их в главе 6, «Основы создания роботов – колеса, питание и электропроводка». Для исследования шагающих движений робота я создал SpiderBot (рис. 1.10) на основе набора для сборки гексапода.

¹ Гексапод – это похожий на паука шестиногий робот, который используется для самых разных целей, в первую очередь связанных с исследованиями в области робототехники. – *Прим. ред.*

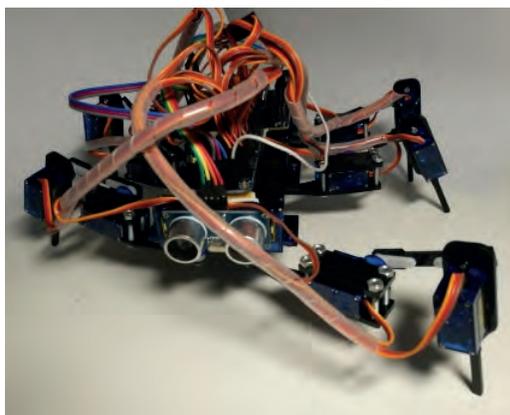


Рис. 1.10. SpiderBot, созданный мной на основе готового проекта. В основе конструкции контроллер ESP8266 и 16-канальный сервоконтроллер Adafruit

Skittlebot был создан мной для участия в Pi Wars 2018. Основой для него послужил переделанный игрушечный экскаватор на дистанционном управлении. **Pi Wars** – это турнир, в котором принимают участие автономные роботы на базе Raspberry Pi, где оценивается их способность выполнять задачи как автономно, так и посредством ручного управления. В нем принимают участие самые разные роботы, и некоторые из них отличаются особым дизайном корпуса или изобретательными инженерными решениями. **Skittlebot** (рис. 1.11) оснащен тремя датчиками расстояния, благодаря которым он умеет обходить стены. Данный тип датчика мы рассмотрим более подробно в главе 8. Также Skittlebot умеет различать цветные объекты посредством камер. Подробнее об этом процессе мы поговорим в главе 13.

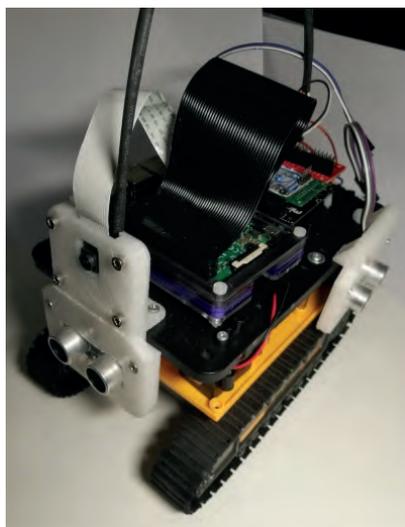


Рис. 1.11. Skittlebot – робот, в основе которого лежит игрушка. Создан мной для участия в Pi Wars 2018

Некоторые любительские роботы создаются с нуля посредством 3D-печати, лазерной резки, вакуумной формовки, обработки дерева, на станках ЧПУ и с помощью других технологий.

Робота, представленного на рис. 1.12, я создал с нуля для лондонской группы робототехников *The Aurorans* в 2009 году. Изначально, он назывался EeeBot, поскольку предназначался для работы под управлением ноутбука Eee PC. Сообщество *The Aurorans* проводило собрания, где участники обсуждали робототехнику. Чуть позже EeeBot был оснащен Raspberry Pi, а также манипулятором (от uArm), благодаря чему он получил название **ArmBot**.

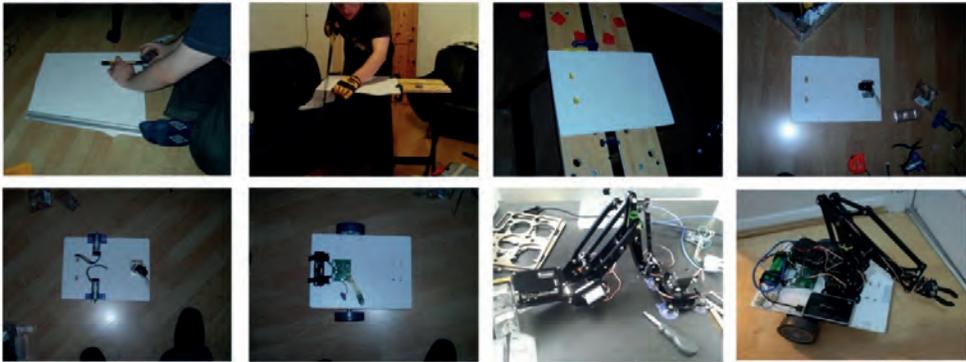


Рис. 1.12. Процесс создания ArmBot

Сегодня на рынке доступно множество готовых наборов для изготовления шасси, поэтому, чтобы создать функционирующего робота, новичку в робототехнике не придется вымерять и вырезать детали. Подобные проекты создаются для проведения экспериментов и призваны побуждать к творчеству других робототехников и детей. Ближе к концу книги мы поговорим о сообществах, которые создают роботов и популяризируют робототехнику. Также мы рассмотрим основные приемы конструирования для создания роботов с нуля.

«Битвы роботов» (*Robot Wars*) – это телевизионное шоу, посвященное соревнованиям между роботами, воплощающими впечатляющие конструкторские и инженерные решения. Однако все роботы-участники управляются вручную, как машинки на дистанционном управлении. Стиральные машины хотя и менее интересны, но гораздо умнее таких роботов.

Выводы

В этой главе мы узнали, что означает слово «*робот*», а также рассмотрели факты и вымыслы, связанные с роботами. Мы определили, что такое настоящий робот и каким требованиям должна отвечать машина, чтобы считаться роботом.

Мы поговорили о роботах, которые встречаются дома и в промышленности. Также вы узнали, что некоторые из них поистине удивительны и предназначаются для полетов на другие планеты. Мы рассмотрели любительских и образовательных роботов и узнали, что некоторые из них созданы просто для развлечения. Вы познакомились с устройством реальных приборов, которых

до этого, возможно, не считали роботами. Мы выяснили, что роботы уже давно проникли в наши дома.

Я надеюсь, что эта глава заставит вас задуматься о том, каким требованиям должен отвечать настоящий робот. Стиральная машина может быть полностью автоматической, запускаться в определенное время по программе, а некоторые современные машины способны уменьшать расход воды путем определения качества и чистоты уже использованной для стирки воды. Однако машины, которые принято называть роботами, на самом деле могут быть просто устройствами на дистанционном управлении. В их числе роботы телеприсутствия и участники «Битв роботов». Несомненно, каждая из таких машин является сложным инженерным продуктом, а для их создания требуются соответствующие высокие навыки.

Одни роботы явно воспринимаются нами как таковые, например ASIMO от Honda и Baxter. В случаях с другими роботами провести черту бывает немного сложнее. Если рассматривать робота как широкую концепцию электро-механической машины, в их ряд не войдут роботы с дистанционным управлением. Если рассматривать их как мобильные машины, то роботом можно будет называть игрушечную машинку на радиоуправлении, но не полностью автономную умную стационарную машину. Машине можно придать человеческие черты. Однако достаточно ли этого для того, чтобы считать ее роботом, если она не сможет выполнять программы и реагировать на окружающую среду?

Теперь, когда мы узнали, что такое робот, предлагаю перейти к следующей главе, в которой мы рассмотрим, как спроектировать робота, чтобы в дальнейшем построить его.

ЗАДАНИЕ

Осмотрите вокруг. Скорее всего, в вашем доме найдется множество автоматических машин, выполняющих функции роботов. Возьмите любой бытовой прибор (кроме стиральной машины) и определите его входы и выходы. Затем визуализируйте их связь с контроллером в виде диаграммы. Подумайте, могут ли эти устройства передвигаться по дому, и если да, то каким образом. Затем выясните, какие петли обратной связи могут присутствовать в этой системе. А что насчет управления этими устройствами? Как оно реагирует на команды пользователя?

ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ

С дополнительными материалами вы можете ознакомиться по следующим ссылкам.

- ASIMO от Honda: <http://asimo.honda.com/>.
- Baxter от Rethink Robotics: <https://www.rethinkrobotics.com/baxter/>.
- Kismet от MIT:
<http://www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html>.
- OhBot: <http://www.ohbot.co.uk/>.

- Марсианская научная лаборатория (The Mars Science Laboratory) от NASA: <https://mars.nasa.gov/msl/>.
- Для создания робота-манипулятора, подобного тому, что используется в ArmBot, ознакомьтесь с MeArm: <https://github.com/mimeindustries/MeArm>.
- Узнать больше о моем ArmBot вы можете по адресу: https://www.youtube.com/watch?v=xY6Oc4_jdmU.