

Practical Robotics in C++

*Build and Program Real Autonomous
Robots Using Raspberry Pi*

Lloyd Brombach



www.bpbonline.com

Ллойд Бромбах

ПРАКТИЧЕСКАЯ РОБОТОТЕХНИКА. C++ и RASPBERRY PI

Санкт-Петербург

«БХВ-Петербург»

2023

УДК 004.4
ББК 32.973.26-018
Б88

Бромбах Л.

Б88 Практическая робототехника. C++ и Raspberry Pi: Пер. с англ. — СПб.: БХВ-Петербург, 2023. — 448 с.: ил.

ISBN 978-5-9775-1200-8

Рассказано о технологии создания автономных роботов на базе одноплатного компьютера Raspberry Pi и о разработке программ для них на языке C++. Показаны принципы написания и даны примеры кода для контроллера привода двигателя, продемонстрированы способы использования датчиков для обнаружения препятствий и построения карт на основе данных лидара. Описаны методы разработки собственных алгоритмов автономного планирования траектории движения, приведен код для автоматической отправки путевых точек контроллеру привода. Рассмотрены библиотеки C++ для написания программ картографии и навигации автономных роботов, даны сведения об использовании контактов аппаратного интерфейса Raspberry Pi GPIO.

Электронный архив на сайте издательства содержит код описанных в книге программ.

Для интересующихся робототехникой

УДК 004.4
ББК 32.973.26-018

Группа подготовки издания:

Руководитель проекта	<i>Павел Шалин</i>
Зав. редакцией	<i>Людмила Гауль</i>
Перевод с английского	<i>Владимира Мелькина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Оформление обложки	<i>Зои Канторович</i>

Copyright 2021 BPB Publications, India. All rights reserved.
First published in the English language under the title *Practical Robotics in C++*,
ISBN 9789389423464 by BPB Publications India (sales@bpbonline.com)
Russian translation rights arranged with BPB Publications, India

© 2021 BPB Publications, Индия. Все права защищены.
Впервые опубликовано на английском языке под названием *Practical Robotics in C++*,
ISBN 9789389423464 издательством BPB Publications India (sales@bpbonline.com)
Права на перевод на русский язык предоставлены издательством BPB Publications, Индия

Подписано в печать 04.04.23.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 36,12.
Тираж 1200 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20
Отпечатано с готового оригинал-макета
ООО "Принт-М", 142300, М.О., г. Чехов, ул. Полиграфистов, д. 1

ISBN 978-93-89423-46-4 (англ.)
ISBN 978-5-9775-1200-8 (рус.)

© BPB Publications, India, 2021
© Перевод на русский язык, оформление. ООО "БХВ-Петербург",
ООО "БХВ", 2023

Оглавление

Об авторе.....	16
Благодарности.....	17
Предисловие	18
Вступительное слово.....	21
Введение	23
ЧАСТЬ I. ВВЕДЕНИЕ В КОМПЬЮТЕРЫ ДЛЯ РОБОТОТЕХНИКИ.....	25
Глава 1. Выбор и настройка компьютера для робота	27
Что такое Raspberry Pi?	27
В чем же разница?	28
Значит, Raspberry Pi — единственный вариант контроллера для управления роботом?	28
Разве Raspberry Pi не предназначен для школ, энтузиастов-электронщиков и игрушек? Я хотел узнать о настоящей робототехнике.	29
Какие модели Raspberry Pi существуют и почему не все из них подходят для наших целей?.....	29
Raspberry Pi Zero и Raspberry Pi ZeroW	31
Raspberry Pi 2B.....	31
Raspberry Pi 3B — лучший выбор!.....	32
Raspberry Pi 3B+.....	32
Новая модель Raspberry Pi 4	33
Выбор операционной системы	33
Raspbian	34
Ubuntu	34
Установка и настройка операционной системы	35
Установка полной Ubuntu Desktop на ноутбук или настольный ПК	36
Установка Lubuntu на Raspberry Pi	36
Установка и настройка интегрированной среды разработки (IDE).....	40
Visual studio Code для ноутбука или настольного ПК.....	40
Code Blocks для Raspberry Pi	41

Заключение.....	42
Вопросы.....	42
Глава 2. Назначение и использование контактов интерфейса GPIO	43
Общие сведения о GPIO.....	43
Что такое интерфейс GPIO	43
Какие же именно функции выполняет GPIO?.....	45
Электроника для программистов	45
Типы выходных данных.....	50
Типы входных данных	52
Некоторые распространенные радиодетали.....	53
Контакты GPIO в качестве выходов	57
Две системы нумерации контактов GPIO.....	58
Контакты GPIO в качестве входов	59
Как получить доступ к контактам GPIO Raspberry Pi с помощью программ на C++	61
Библиотека I2C	62
Установка и настройка библиотеки I2C	62
Удостоверимся, что Code::Blocks может обращаться к I2C	63
Запуск программ I2C	64
Наш первый проект — hello_blink	64
Цифровой вход, управляющий цифровым выходом — hello_button.....	67
Функции обратного вызова для обработки событий GPIO.....	68
Заклучение.....	71
ЧАСТЬ II. НАЧИНАЕМ ПРОЕКТИРОВАТЬ РОБОТА	73
Глава 3. Платформа для робота.....	75
Общие сведения.....	75
Габаритные размеры и режим эксплуатации	76
Что лучше — дифференциальный рулевой привод или привод Аккермана?.....	78
Дифференциальный рулевой привод	78
Рулевой привод Аккермана	79
Готовые платформы для роботов.....	79
Большие готовые платформы	79
Маленькие готовые роботы	80
Советы по созданию собственного робота.....	82
Материалы для конструирования.....	82
Аккумуляторы.....	83
Ходовая часть.....	83
Где найти детали для роботов	84
Перепрофилирование роботов-пылесосов или автомобилей с дистанционным управлением	85
Роботы-пылесосы с интерфейсом	86
Взаимодействие с Roomba	87
"Разморозка" Roomba	90
Роботы-пылесосы без интерфейса	91
Перепрофилирование автомобилей и грузовиков с дистанционным управлением	92
Заклучение.....	94
Вопросы.....	94

Глава 4. Типы двигателей для роботов и управление двигателями	95
Общие сведения	95
Типы двигателей	95
Сравнение двигателей переменного (AC) и постоянного (DC) токов	96
Щеточные двигатели постоянного тока	97
Сервоприводы	98
Шаговые двигатели	99
Бесщеточные двигатели постоянного тока (BLDC)	100
Принципы работы транзистора и контроллеры двигателей	100
Простейший способ управления: включено/выключено	101
Транзисторы	102
Широтно-импульсная модуляция (ШИМ)	105
ШИМ для создания аналоговых напряжений	105
ШИМ в качестве управляющего сигнала	106
Драйверы и контроллеры двигателей	107
Драйверы двигателей	108
Управление двигателями с помощью драйвера двигателя на основе двойного H-моста L298N	109
Контроллеры двигателей	112
Заключение	113
Вопросы	114
Бонусное задание	114
Глава 5. Связь с датчиками и другими устройствами	115
Общие сведения	115
Двоичные (логические) сигналы	115
Выключатели с защитой от дребезга контактов	116
Колесные энкодеры	117
Двоичные сигналы от аналоговых датчиков	118
Сводная информация о передаче данных с помощью двоичных сигналов	119
Передача данных через последовательный интерфейс	119
Последовательная передача данных с помощью UART	119
Настройка Raspberry Pi и тестирование последовательной передачи данных через UART	121
Устранение ошибки при открытии последовательного порта	124
Передача данных через последовательную шину I2C	124
Настройка и использование устройства I2C с Raspberry Pi	126
Пример и тестовая программа: <i>hello_i2c_lsm303</i>	127
Заключение	130
Вопросы	131
Глава 6. Дополнительное оборудование	132
Общие сведения	132
Источники питания	132
Источники питания напряжением 5 В	133
Регулируемые источники питания	133
Релейные блоки	134
Преобразователи логических уровней	135
Преобразователи интерфейса (FTDI)	136
Микроконтроллеры Arduino	137
Микроконтроллеры Digispark	137

Заключение.....	138
Вопросы.....	138
Глава 7. Установка компьютера, управляющего роботом	139
Общие сведения.....	139
Последовательность шагов.....	140
Установка компьютера и подача питания на него.....	140
Соединение компьютера с остальными частями робота.....	141
Заключение.....	143
Вопросы.....	144
ЧАСТЬ III. ЛОГИКА ФУНКЦИОНИРОВАНИЯ РОБОТА	145
Глава 8. Стратегия управления роботом.....	147
Общие сведения.....	147
Управление роботом: верхний и нижний уровни.....	147
Основной контур управления.....	149
Наблюдение и сравнение.....	149
Реагирование.....	150
Воздействие.....	150
Контроллеры с разомкнутым и замкнутым контуром управления.....	153
Разработка контроллеров верхнего уровня (главных контроллеров).....	154
Разработка контроллеров нижнего уровня (технологических контроллеров).....	156
Двухпозиционные контроллеры (регуляторы типа включено-выключено).....	157
Пропорциональные контроллеры.....	157
Проектирование контроллеров, допускающих некоторую погрешность.....	161
Установка минимального значения выходного сигнала.....	161
За рамками пропорциональных контроллеров.....	162
Заключение.....	163
Вопросы.....	163
Глава 9. Организация совместной работы компонентов	164
Общие сведения.....	164
Что такое операционная система для роботов?.....	165
ROS или создание собственного ПО для управления роботами?.....	165
ROS и область коммерческой робототехники.....	166
Установка ROS.....	167
Установка ROS Melodic на ноутбуке или настольном компьютере.....	167
Установка ROS Kinetic на Raspberry Pi 3B.....	168
Быстрое тестирование ROS.....	170
Краткий экскурс в ROS.....	171
Пакеты, узлы, издатели, подписчики, топики и сообщения.....	171
Полезные приемы.....	177
Создание и написание пакетов и узлов ROS.....	178
Файловая система ROS.....	178
Создание пакетов ROS.....	178
Написание программ ROS (узлов).....	180
Загрузка, просмотр и запуск программ, скачанных для этой главы.....	186
Как облегчить жизнь с помощью файлов <i>roslaunch</i> и <i>.launch</i>	187
Заключение.....	188
Вопросы.....	189

Глава 10. Карты для определения местоположения робота	190
Общие сведения	190
Угол, курс, расстояние: общепринятые соглашения	191
Получение данных с датчиков	193
Сетчатая карта занятости	194
Построение сетчатых карт занятости (OGM) с помощью данных, получаемых с датчиков.....	196
Маркировка занятых ячеек	199
Маркировка свободных ячеек	202
Заключительные шаги при составлении карты.....	202
Публикация карты в виде сообщения ROS	203
Преобразования в ROS.....	204
Для чего нужны преобразования.....	205
Использование преобразований в ROS.....	206
Публикация преобразований с помощью <i>static transform publisher</i>	207
Публикация сообщений от узлов с помощью транслятора преобразований (<i>transform broadcaster</i>).....	208
Получение данных о преобразованиях в узлах.....	209
Просмотр данных о преобразовании из командной строки.....	211
Упрощение картографирования с помощью Gmapping	211
Общие сведения о Gmapping	212
Установка Gmapping.....	212
Запуск Gmapping и задание параметров в файлах запуска	212
Этапы создания карты.....	214
Визуализация создания карты в реальном времени	214
Сохранение карты и ее последующее использование.....	216
Сохранение карт	216
Загрузка ранее сохраненной карты	217
Заключение.....	217
Вопросы.....	217
Глава 11. Отслеживание перемещений и локализация робота	219
Общие сведения	219
Поза робота	220
Преобразование углов Эйлера в кватернионы	221
Преобразование кватернионов в углы Эйлера	223
Одометрия и точный расчет траектории	223
Колесная одометрия	224
Расчет расстояния, пройденного каждым колесом.....	227
Расчет общего расстояния, пройденного роботом.....	228
Расчет изменения угла поворота тета	228
Сложение величины изменения угла поворота с предыдущим значением угла поворота тета.....	229
Расчет расстояния, пройденного по осям x и y (преобразование координат)	229
Прибавление полученных расстояний к соответствующим значениям предыдущей оценки позы.....	230
Публикация сообщения одометрии о новой позе для других узлов	230
Сохранение данных новой позы для использования в следующем цикле	230
Точный расчет траектории.....	230

Публикация данных одометрии в ROS.....	232
Издатель сообщений о преобразованиях одометрии.....	234
Дальнейшее отслеживание перемещений робота и его локализация в пространстве.....	236
Инструмент для коррекции позы вручную.....	236
Фидуциальные маркеры.....	237
Локализация с помощью лазерного сканера.....	238
GPS и GNSS.....	239
Системы локализации на основе радиомаяков.....	239
Заключение.....	240
Вопросы.....	240
Глава 12. Автономное движение.....	241
Общие сведения.....	241
Обзор движения роботов в ROS.....	241
Контроллер двигателя — <code>simple_diff_drive.cpp</code>	242
Шаги по созданию контроллера двигателя <code>simple_diff_drive</code>	243
Код контроллера двигателя дифференциального рулевого привода, представленный в общем виде.....	244
Код контроллера двигателя дифференциального рулевого привода.....	245
Контроллер привода — <code>simple_drive_controller.cpp</code>	251
Шаги по созданию контроллера привода.....	251
Заключение.....	256
Вопросы.....	256
Глава 13. Автономное планирование маршрута.....	257
Общие сведения.....	257
Методы планирования маршрута и сопутствующие проблемы.....	257
Проблемы.....	258
Методы планирования маршрута.....	258
Увеличение границ вокруг препятствий.....	259
Карты затрат (<code>costmap</code>).....	260
Пакет <code>costmap_2d</code>	260
Планирование маршрута с помощью алгоритма A*.....	263
Как работает алгоритм A*.....	264
Пошаговый разбор алгоритма A*.....	266
Разбор процедуры A*.....	268
Написание программы A* как узла ROS.....	273
Стандартные вещи, вспомогательные функции и <code>main()</code>	274
Сердце узла A*: функция <code>find_path()</code>	284
Заключение.....	289
Вопросы.....	289
ЧАСТЬ IV. ИНТЕРПРЕТАЦИЯ ДАННЫХ, ПОСТУПАЮЩИХ С ДАТЧИКОВ.....	291
Глава 14. Колесные энкодеры для одометрии.....	293
Общие сведения.....	293
Колесные энкодеры.....	293
Оптические энкодеры.....	294
Энкодеры на датчиках Холла.....	294

Подключение энкодеров	295
Издатель сообщений об импульсах: <code>tick_publisher.cpp</code>	297
Код издателя сообщений об импульсах энкодера	298
Заключение	302
Вопросы	302
Глава 15. Ультразвуковые датчики расстояния	303
Общие сведения	303
Основная информация об ультразвуковом дальнмере HC-SR04	304
Считывание показаний HC-SR04	304
Подключение HC-SR04	304
Издатель данных ультразвукового измерения расстояния: <code>ultrasonic_publisher.cpp</code>	305
Издатель сообщений об ультразвуковом измерении расстояния:	
пошаговый разбор	305
Обзор кода издателя сообщений об ультразвуковом измерении расстояния	306
Использование данных ультразвукового измерения расстояния при обнаружении объектов	309
Заключение	310
Вопросы	311
Глава 16. Инерциальные измерительные блоки (IMU) — акселерометры, гироскопы и магнитометры	312
Общие сведения	312
Акселерометры	313
Недостатки акселерометра	314
Публикация данных IMU в ROS	314
Тип данных <code>sensor_msgs::Imu</code> в ROS	315
Код издателя сообщений IMU	316
Гироскопы	320
Недостатки гироскопа	321
Добавление данных гироскопа в узел IMU	321
Магнитометры	322
Недостатки магнитометра	322
Добавление данных магнитометра	323
Установка IMU	325
Заключение	325
Вопросы	326
Глава 17. GPS и системы на основе внешних радиомаяков	327
Общие сведения	327
Как работают системы на основе радиомаяков	327
Основные сведения о GPS и GNSS	329
Точность GPS/GNSS	329
Определение местоположения с помощью GPS/GNSS-RTK с точностью до 2 см	330
Ограничения GPS/GNSS	331
Данные GPS/GNSS	332
Строки с данными NMEA	332
Некоторые основные представления данных о широте и долготе	334
Публикация данных GPS/GNSS в ROS	335
Пакет ROS: <code>nmea_navsat_driver</code>	335

Установка пакета <i>nmea_navsat_driver</i>	336
Изучение документации к пакетам ROS	337
Запуск узла <i>nmea_serial_driver</i> с параметрами	338
Заключение.....	339
Вопросы.....	339
Глава 18. Устройства LIDAR и данные, которые они предоставляют	340
Общие сведения.....	340
Основные сведения об устройствах LIDAR.....	340
Ограничения LIDAR	341
Типы LIDAR	342
Однонаправленный (одноточечный) LIDAR	342
2D-LIDAR.....	343
3D-LIDAR.....	344
LIDAR, установленный на роботе-пылесосе	344
Критерии выбора LIDAR	346
Данные LIDAR: сообщение <i>sensor_msgs::LaserScan</i>	347
Факторы, которые необходимо учитывать при монтаже устройств LIDAR.....	349
Установка, запуск и испытание распространенной модели LIDAR	350
Действия по настройке RPLIDAR	351
Визуализация сообщения <i>LaserScan</i>	353
Заключение.....	356
Вопросы.....	356
Глава 19. Реальное зрение с помощью видеокамер.....	357
Общие сведения.....	357
Что такое изображение?.....	358
Атрибуты изображения	359
Координаты пиксела.....	359
Проверка наличия или установка необходимого программного обеспечения	360
ROS Kinetic.....	360
ROS Melodic	361
Тестирование OpenCV в ROS.....	362
Программное обеспечение для обработки изображений (OpenCV) и ROS	363
Шаг 1. Публикация изображений в ROS	364
Установка <i>usb_cam_node</i>	364
Запуск <i>usb_cam_node</i>	364
Тестирование выходного сигнала камеры.....	366
Шаг 2. Подпишитесь на сообщение об изображении в другом узле	367
Создайте свой пакет ROS для видеонаблюдения	367
Написание кода для подписчика на сообщения с изображением	368
Шаг 3. С помощью <i>cv-bridge</i> преобразуйте изображение RGB, которое использует ROS, в изображение BGR, с которым может работать OpenCV	369
Шаг 4. Выполните необходимые операции с изображением	369
Шаг 5. Опубликуйте любые данные, не относящиеся к изображению, в виде отдельного сообщения ROS	370
Шаг 6. Преобразуйте измененное изображение обратно в формат RGB	370
Шаг 7. Опубликуйте итоговое изображение в отдельном топике	371
Еще немного про обработку изображений.....	371
Ядра, диафрагмы и блоки	371

Важность работы с копиями вместо оригинальных изображений.....	372
Несколько слов об освещении.....	373
Ревизия шага 4 с включением большого количества операций OpenCV	373
Преобразование цветового формата: <i>cvtColor()</i>	374
Размытие изображений: <i>blur()</i> , <i>medianBlur()</i> , <i>GaussianBlur()</i>	374
Выделение краев: <i>Canny()</i>	375
От определения краев на изображении к числовым значениям: <i>HoughLinesP()</i>	376
Маскирование изображения: <i>bitwise_and()</i>	380
Фильтрация по цвету: <i>cvtColor()</i> и <i>inRange()</i>	383
Полезные инструменты ROS.....	387
Расширенные функции OpenCV и не только	387
Распознавание изображений с помощью облачных технологий.....	388
Заключение.....	389
Вопросы.....	389
Глава 20. Совместное использование различных датчиков.....	390
Общие сведения.....	390
Доступно о совместном использовании датчиков.....	391
Датчик абсолютной ориентации Bosch BN0055	391
Предоставляемые данные	392
Улучшенная одометрия.....	392
Интеграция BN0055: оборудование и издатель ROS	393
Интеграция BN0055: узел одометрии	394
Шаг 1. Подписать на сообщение IMU.....	395
Шаг 2. Проверить, что поле ориентации не помечено как "do not use" (не использовать).....	395
Шаг 3. Преобразовать кватернионы в углы Эйлера	396
Шаг 4. Сохранить информацию о смещении, если это первое сообщение IMU	396
Шаг 5.1. Если это НЕ первое сообщение IMU, сохранить курс IMU	397
Шаг 5.2. Передать новое значение курса IMU в функцию расчета параметров одометрии.....	397
Комплексный подход к совместному использованию датчиков.....	398
Фильтр Калмана.....	398
Ковариационная матрица.....	401
Ковариационные матрицы в сообщениях ROS	402
Узел <i>robot_pose_ekf</i>	403
Установка <i>robot_pose_ekf</i>	404
Запуск <i>robot_pose_ekf</i>	404
Последнее замечание о преобразованиях и <i>roslaunch</i>	405
Заключение.....	406
Вопросы.....	406
ЧАСТЬ V. РАЗРАБОТКА АВТОНОМНОГО РОБОТА	407
Глава 21. Сборка и программирование автономного робота	409
Общие сведения.....	409
Раздел 1. Создание физической платформы для робота	410
Платформа для робота: общий обзор и список деталей.....	411
Модули, объединяющие в себе колесо и двигатель	413
Драйвер(ы) двигателя.....	414

Ролик (третье колесо робота).....	414
Аккумуляторы и зарядное устройство.....	414
Шасси/основание.....	415
Компьютеры.....	416
LIDAR или другой датчик расстояния.....	416
Колесные энкодеры.....	417
IMU (инерциальный измерительный блок).....	417
Преобразователь напряжения для компьютера.....	417
Коммутационная плата для колодки GPIO.....	418
Видеокамера.....	418
Вольтметр для контроля напряжения аккумулятора.....	419
Различные материалы.....	420
Сборка платформы для робота.....	420
Подготовьте компьютер.....	421
Подготовьте колесные модули.....	421
Продумайте схему расположения компонентов.....	422
Подготовьте шасси.....	422
Установите колесные модули и ролик.....	422
Установите драйвер двигателя, клеммные колодки и источник питания для компьютера.....	424
Подготовьте коммутационную плату для колодки GPIO.....	424
Установите компьютер, коммутационную плату для колодки GPIO и IMU.....	424
Соедините все блоки проводами и установите аккумулятор.....	425
Установите LIDAR и видеокамеру.....	425
Еще несколько советов.....	427
Раздел 2. Программирование робота.....	428
Программирование: общие замечания.....	428
Программирование робота: подробная инструкция.....	429
1. Создать папку проекта.....	430
2. Получить данные датчиков для публикации.....	430
3. Настроить управление платформой с помощью пульта ДУ.....	432
4. Отслеживать перемещение робота и публиковать данные о местоположении.....	435
5. Обеспечить перемещение робота по путевым точкам (без обхода препятствий).....	436
6. Составьте карту окружения робота.....	438
7. Загрузить сохраненную карту с помощью файлов запуска.....	439
8. Добиться автономной навигации робота в пределах карты.....	440
Запуск автономного робота!.....	441
Некоторые советы по устранению неполадок.....	441
Что дальше?.....	442
Обход динамических препятствий.....	442
ПИД-регуляторы.....	443
Главный контроллер, который управляет различными процедурами или задачами.....	443
Реализация преобразования <code>map</code> в <code>odom</code> (полная локализация).....	443
Следите за новостями в Интернете.....	443
Заключение.....	444
Приложение. Описание электронного архива.....	445
Предметный указатель.....	446

*Посвящается
инженерам, которые создавали, создают
и будут создавать роботов*

Об авторе

Ллойд Бромбах живет в Рочестер-Хиллз, штат Мичиган, США. Он является вторым капитаном команды клуба "Wayne Robotics Club" на соревнованиях по робототехнике. Занимается решением задач, связанных с робототехническими системами. Начиная с участия в конкурсе НАСА 2007 года по созданию автономного робота для раскопки лунного грунта. В 2019 году принимал участие в конкурсе по интеллектуальным наземным транспортным средствам, работая в составе команды разработчиков программного обеспечения для планирования маршрута и обхода препятствий.

Инженер по системам управления с 15-летним опытом работы. Прошел двухгодичную программу обучения электронике, теории автоматического управления, датчикам и микроконтроллерам, а также четырехлетнюю программу стажировки инженеров. Применяет полученные знания при разработке и внедрении специализированных электромеханических решений и наставничестве новых членов клуба робототехники. Будучи твердо убежденным в необходимости приобретения разносторонних современных навыков, он вновь вернулся на университетскую скамью, получив в этот раз диплом о высшем образовании по компьютерным наукам.

Благодарности

В процессе написания этой книги я получил столько слов поддержки, понимания и ободрения, что трудно решить, с чего начать.

Прежде всего, я должен поблагодарить мою маму. Это она убедила меня в детстве, что все в моих силах. Моего отца, который показал мне отправную точку. Моих коллег и в особенности Мо — за то, что они приходили мне на помощь, когда дел становилось слишком много. Моих друзей и семью, которые обходились без меня на большинстве встреч в последние годы, потому что я должен был исследовать, работать, составлять графики (чего греха таить — я самый одержимый из нашей компании). Благодарю доктора Пандью за его наставничество, а также членов клуба робототехники Университета Уэйна и команду, участвующую в конкурсе по интеллектуальному наземному транспорту, за приобретение опыта обучения и преподавания. Выражаю особую признательность сотрудникам издательства BPB — за то, что они убедили меня взяться за написание книги и оставались со мной до конца.

Я благодарю свою жену Джен за ее невероятное терпение, понимание и поддержку. За то, что она стойко держалась и даже готовила еду для команды Wayne Robotics, когда мы засиживались допоздна. За то, что мирилась с моими отлучками, когда я по 14 часов пропадал на конференциях, собраниях команды разработчиков или же просто работал в офисе или в подвале нашего дома. За то, что ни разу не пожаловалась, когда я брал роботов с собой на отдых, и когда она спотыкалась о них по дому.

Благодарю моих читателей за то, что они увлечены захватывающим и сложным миром робототехники. За их энтузиазм, а также за понимание того обстоятельства, что, несмотря на все старания автора, в каждой работе могут быть недостатки. Я буду признателен за отзывы и замечания, чтобы читатели последующих версий книги и интернет-сообщество Practical Robotics не оказались разочарованы.

Я выражаю слова благодарности этим и многим другим людям, которые поддержали и направили меня, сделав возможным написание этой книги.

Спасибо вам!

Предисловие

Найти четкое руководство по созданию автономного робота — непростая задача. Дело в том, что для этого требуются сведения из множества разных дисциплин, а существующие ресурсы, как правило, посвящены какой-то одной области. В книгах по электронике не описывается картографирование или планирование маршрута робота, а в книгах по алгоритмам робототехники не приводится теория автоматического управления, необходимая для взаимодействия программного обеспечения с двигателями и датчиками обратной связи. И кроме того, во многих книгах и учебниках по различным предметам предполагается, что читатель уже имеет ученую степень или, по крайней мере, знаком с высшей математикой.

А теперь — хорошие новости. Благодаря этой книге вам не придется тратить годы, изучая те дисциплины, которые не понадобятся для создания и программирования вашего первого автономного робота. Наоборот, она даст вам четкий план и прочную основу, чтобы вы смогли быстрее войти в курс дела. Я также подскажу вам, что конкретно нужно изучать далее по каждой из дисциплин, чтобы вы могли двигаться в тех направлениях, которые покажутся вам наиболее интересными, не распыляя внимание на второстепенные вещи.

Чего я не предоставлю, так это руководств, которым можно слепо следовать без понимания сути. Хотя весь программный код приведен здесь в печатном виде и доступен для загрузки на github.com/lbrombach, моя цель не в том, чтобы вы построили и запрограммировали точно такого же робота, как в примере. Вместо этого я призываю вас проектировать, создавать и программировать роботов по-своему, опираясь на приведенные примеры. Я дам вам хороший первоначальный толчок, но ваше дальнейшее продвижение зависит лишь от вашей фантазии и целеустремленности.

Эту книгу я писал, работая одновременно со студентами университета, опыт и компетенции которых сильно различались. Мне приходилось обучать как первокурсников, не имеющих никакого опыта, так и аспирантов с дипломами в области информатики, электротехники и машиностроения. Никто из них не умел создавать и программировать роботов. Лучшие программисты не разбирались в двигателях и датчиках, а инженеры-электротехники не знали, как превратить данные, посту-

пающие с датчиков, в полезные значения кода программы. Каждый семестр мы, капитаны команды, обнаруживали, что снова объясняем базовые вещи по каждому из предметов, и мечтали о едином пособии, которое могло бы дать всем членам нашей команды всеобъемлющую основу. В частности, нам нужны были уроки, в меру простые для новичков, но при этом достаточно обстоятельные, чтобы и выпускники могли извлечь из них пользу. Вы держите в руках именно такое руководство, и для начала работы с ним не требуется ничего, кроме базовых знаний C++ и желания учиться.

Эта книга будет интересна как начинающим, так и самым опытным программистам. Я создавал программный код, который, мне кажется, будет легко понять начинающему разработчику. При этом у более опытных программистов не возникнет проблем с реализацией собственных идей. Информация для начинающих программистов: в данной книге работа алгоритмов демонстрируется с помощью синтаксиса языка C++, но не думайте, что это "книга по программированию". По мере совершенствования навыков программирования вы будете замечать все больше мест, где я избегал применения тех функций C++, которые, как мне казалось, будут трудны для новичков, используя, например, глобальные переменные, чтобы сделать код более понятным. Все приведенные примеры программ работоспособны, но могут оказаться не вполне оптимальными. Мне совсем не хотелось перегружать книгу сложным программным кодом. Когда по ходу книги конструкции на C++ неизбежно становятся чуть более сложными, я предупреждаю об этом и даю краткие пояснения.

Роботы для служб доставки, для складов, для использования в быту, любительские и сельскохозяйственные роботы — у каждого из них собственное предназначение и свой режим эксплуатации. Однако в их основе лежат одни и те же базовые принципы, о которых вы узнаете далее. Я советую не спешить с чтением этой книги. Осваивайтесь потихоньку и проходите учебник до конца, после чего возвращайтесь к практике.

Не терпится начать? Ну что же, сейчас самое время приступить к делу.

Материал книги разбит на пять частей, в которых представлены следующие темы.

Часть I "Введение в компьютеры для робототехники" содержит две главы. Первая посвящена выбору и настройке ноутбуков и компьютеров Raspberry Pi. Во второй главе подробно рассказывается о контактах аппаратного интерфейса ввода/вывода общего назначения (GPIO) Raspberry Pi, приводятся начальные сведения по электронике и использованию программ для взаимодействия с аппаратной составляющей.

Часть II озаглавлена "Начинаем проектировать робота". Здесь собрана информация, необходимая для создания платформы для робота, которая затем может быть запрограммирована на автономное поведение в последующих разделах. В пяти главах части обсуждаются типы доступных платформ для роботов, приобретение или создание двигателей определенного типа и управление ими, связь между различными узлами робота, всевозможное вспомогательное оборудование, подключение компьютера и остальных элементов робота.

Часть III содержит шесть глав, посвященных вопросам логики функционирования роботов. В первой главе рассматриваются контроллеры — как для отдельных процессов (например, одного двигателя), так и для сложной совокупности систем. Во второй главе речь идет о согласованном взаимодействии десятков процессов и огромного объема данных, необходимых для автономного робота. Оставшиеся четыре главы этой части посвящены основным вопросам, связанным с автономными роботами: как заставить робота двигаться самостоятельно, как составить карту для навигации, как робот может определять пройденное расстояние и самостоятельно находить путь, избегая препятствий.

Часть IV называется "Интерпретация данных, поступающих с датчиков". Здесь подробно рассматриваются датчики, полезные в вышеперечисленных случаях, и некоторая другая информация. Пять глав из семи посвящены датчикам для определения расстояния, пройденного роботом (одометрия), обнаружению препятствий с помощью ультразвуковых датчиков, лазерного дальномера (LIDAR), GPS и других систем определения местоположения на основе радиомаяков. Одна глава посвящена объединению данных с нескольких датчиков для повышения надежности системы, а еще одна — использованию видеокамер для компьютерного зрения.

Часть V "Разработка автономного робота" содержит одну заключительную главу, которая подытоживает все материалы, касающиеся создания и программирования автономного робота, рассмотренные в предыдущих главах.

Вступительное слово

Радостное настроение царит в тех областях науки и техники, где сходятся воедино робототехника, микропроцессоры, информационные технологии и передовые алгоритмы управления автономными роботами. Эта книга выходит в переломный момент развития технологий, когда они начинают ориентироваться на робототехнику. Роботы уже играют приоритетную роль в таких областях, как мобильные системы, производство, сельское хозяйство, здравоохранение и управление логистическими цепочками. Однако, по некоторым оценкам, ежегодный рост на рынке промышленной робототехники составит почти 12%, а объем рынка в следующем десятилетии составит более 33 млрд. долларов. При этом размер рынка медицинской робототехники достигнет 100 млрд долларов к 2024 году. Грядет робототехническая революция, которая поможет нам решить многие важнейшие проблемы как в настоящем, так и в будущем.

Робототехника уже интегрирована во многие аспекты нашей экономики и повседневной жизни. И этот процесс продолжает набирать обороты. Заказываете что-то на Amazon? Робот Kiva находит ваш товар на полке и доставляет его в зону упаковки. Предстоит уборка в доме? На помощь приходит робот-пылесос. По мере роста населения земного шара люди будут сталкиваться с огромными проблемами, и роботы будут все чаще приходить нам на помощь. Они смогут более эффективно вести сельское хозяйство, тушить пожары с земли и с воздуха и даже забирать наших детей с футбольной тренировки.

Но в связи с таким бурным развитием отрасли возникает острая потребность в квалифицированных инженерах и исследователях — сотрудниках, которые понимают, как работают оборудование, интеллектуальные алгоритмы, программное обеспечение и датчики в составе единой системы, и которые могут решать важные и сложные проблемы, связанные с робототехникой. Данная книга сыграет ключевую роль в подготовке этих кадров благодаря практическому подходу к изучению робототехники. За счет широкого охвата тематических областей она увлекает читателей и помогает им лучше понять основы робототехники. В ней раскрываются ключевые аспекты таких объемных тем, как операционная система для роботов (ROS), интеграция с микроконтроллерами (например, Raspberry Pi), применяемые датчики (на-

пример, Lidar, GPS, энкодеры, инерциальный измерительный блок и т.д.) и разработка программного обеспечения.

Отличительной чертой книги "Практическая робототехника на C++" является то, что в ней показано, как объединять различные имеющиеся технологии для создания действующего робота, и в то же время довольно подробно разбираются ключевые алгоритмы (такие как планирование маршрута, автономная навигация и т. д.), что позволяет читателю применить полученные знания практически к любому робототехническому проекту. Книга написана таким образом, что у читателя будет четкое понимание, как создать действующего робота. Более того, она побуждает его продолжать исследования и эксперименты в области робототехники, углубляясь в более сложные разделы.

Я познакомился с Ллойдом Бромбахом, когда мы основали клуб робототехники в Государственном университете Уэйна в Детройте. Я работал куратором на факультете, а Ллойд, студент факультета информатики, был капитаном команды робототехников. Мы приняли участие в конкурсе по интеллектуальному наземному транспорту, и Ллойд курировал разработку основного программного обеспечения для планирования маршрута и обхода препятствий на базе операционной системы для роботов (ROS). Я обратил внимание на его настойчивость и терпение, живой ум и чрезвычайное упорство. Когда мы разговорились, я узнал, что в 2012 году он участвовал в соревнованиях Ironman с суммарной дистанцией 140,6 мили — спортивном состязании, которое требует не только невероятной физической выносливости, но и огромной силы духа. Он привнес в эту книгу ту же настойчивую решимость. Это настоящий боец — не только спортивного, но и научного фронта.

Можно сказать, что он действительно нашел свою стихию.

Абхилаш Пандья,

доктор философии, доцент,

директор факультета электротехники и вычислительной техники.

Лаборатория усовершенствованных интеллектуальных робототехнических систем (CARES), Государственный университет Уэйна, Детройт

Введение

Найти исчерпывающее руководство по созданию автономного робота — непростая задача. Дело в том, что для этого требуются сведения из великого множества дисциплин, а существующие ресурсы, как правило, посвящены какой-то одной области. В книгах по электронике не рассказывается о картографировании или планировании маршрута, а в книгах по алгоритмам робототехники нет сведений из теории автоматического управления, необходимой для взаимодействия программного обеспечения с двигателями и датчиками. И кажется, что во многих книгах и учебниках по всем предметам неявно предполагается, что читатель уже имеет ученую степень или, по крайней мере, знаком с продвинутой математикой.

Понимание робототехники, позволяющее проектировать и конструировать собственных роботов, сложилось у меня, как мозаика из сведений, полученных по разным каналам:

- ◆ Двухгодичный курс по электронике и технологии автоматизации.
- ◆ Два года изучения информатики, программирования и математики.
- ◆ Год изучения датчиков, микроконтроллеров и цифровых коммуникаций.
- ◆ Год изучения программного обеспечения и алгоритмов робототехники.
- ◆ Многолетний опыт проектирования и ремонта автоматизированных систем управления для промышленного и коммерческого оборудования.
- ◆ Все годы моей жизни, которые я посвятил созданию и совершенствованию электронных и электромеханических устройств (прошу прощения у жены за инцидент с роботом-пылесосом).
- ◆ Десятки книг и онлайн-курсов.
- ◆ Десятки своих собственных и совместных робототехнических проектов, включая конкурс НАСА 2007 года по разработке автономного робота для раскопки лунного грунта и конкурс по интеллектуальным наземным транспортным средствам в 2019 году.

Я помогу вам сэкономить четыре года, а то и более, т. к. вам не придется тратить время на изучение тех разделов, которые не нужны для создания и программирова-

ния вашего первого автономного робота. Напротив, эта книга даст вам четкий план действий и заложит прочную базу, чтобы вы могли успешно "включиться в игру". Я также подскажу вам, что изучать дальше по каждой из дисциплин, чтобы вы могли сфокусироваться и двигаться в тех направлениях, которые покажутся вам наиболее интересными.

В этой книге вы не найдете руководство, которым можно слепо следовать без понимания сути. Наоборот, я призываю вас проявлять максимум самостоятельности, проектировать, создавать и программировать роботов по-своему, опираясь на приводимые примеры. В таком случае успех будет зависеть лишь от вашей фантазии и целеустремленности.

Одновременно с написанием книги я работал со студентами университета, опыт и способности которых варьировались в широком диапазоне. Мне приходилось обучать как первокурсников, не имеющих никакого опыта, так и аспирантов с дипломами в области информатики, электротехники и машиностроения. Но никто из них не умел создавать и программировать роботов. Программисты не разбирались в двигателях и датчиках, а инженеры-электротехники не знали, как интерпретировать в программе данные, поступающие с датчиков. Мы, капитаны команды, мечтали о едином пособии, которое могло бы дать всем нашим разработчикам всеобъемлющую основу. Материал книги должен быть понятным для новичков, но при этом достаточно обстоятельным, чтобы и опытные разработчики могли извлечь пользу. Перед вами именно такое руководство, и для начала работы с ним не требуется ничего, кроме базовых знаний C++ и желания учиться.

Я старался написать книгу, которая будет интересна как начинающим, так и опытным программистам. Программный код будет легко понять начинающему разработчику, а более опытные программисты смогут внедрить собственные наработки. Программный код примеров написан на языке C++. При этом я старался избегать применения тех функций C++, которые будут трудны для новичков, используя, например, глобальные переменные, чтобы сделать код более понятным. Все программы рабочие, но могут оказаться не вполне оптимальными. Мне очень не хотелось перегружать книгу программным кодом. По мере появления все более усложняющихся конструкций на C++ я предупреждаю об этом и даю краткие пояснения.

Несмотря на самые разнообразные области применения роботов в основе их функционирования лежат общие базовые принципы. Рекомендую вам последовательно раздел за разделом изучать теоретический материал и после этого использовать полученные знания на практике.

Давайте же приступим к делу.

Ллойд

ЧАСТЬ I

**Введение в компьютеры
для робототехники**

ГЛАВА 1

Выбор и настройка компьютера для работа

В этой главе мы рассмотрим следующие темы:

- ◆ Что такое Raspberry Pi?
- ◆ Какие модели Raspberry Pi существуют и почему не все из них подходят для наших целей?
- ◆ Выбор операционной системы.
- ◆ Установка и настройка операционной системы.
- ◆ Установка и настройка интегрированной среды разработки (IDE).

Что такое Raspberry Pi?

Raspberry Pi — это небольшой одноплатный компьютер (Single-Board Computer, SBC). В отличие от вашего домашнего настольного ПК, содержащего системную плату со слотами или разъемами для процессора, памяти, видеопроцессора и других компонентов, которые вы можете менять и настраивать по своему усмотрению, при производстве SBC все эти элементы устанавливаются на одной печатной плате. Их невозможно поменять или модернизировать без замены всей платы. В одноплатном компьютере нет жесткого диска, а операционная система запускается с карты памяти microSD. На сегодняшний день существует целый ряд моделей SBC, но Raspberry Pi, несомненно, наиболее популярен, поскольку очень дешев и имеет огромную поддержку со стороны сообщества. Если у вас возникнет какая-либо проблема, многие тысячи энтузиастов Raspberry Pi готовы будут прийти вам на помощь в группах социальных сетей или на официальных форумах компании Raspberry Pi Foundation по адресу www.raspberrypi.org.

Несмотря на свои компактные габариты (самая большая модель — размером с колоду карт), одноплатные компьютеры Raspberry Pi являются полнофункциональными компьютерами под управлением полноценной операционной системы (обычно Linux) и способны делать многое из того, что и ваш настольный ПК. Например, они позволяют просматривать веб-страницы, воспроизводить видео, отправлять электронную почту, к ним можно подключать принтеры, видекамеры и т. д. Кроме

того, они способны выполнять операции, которые настольный ПК не может делать без оснащения дополнительным оборудованием, например могут напрямую считывать данные с кнопок, датчиков или напрямую взаимодействовать с наиболее распространенными контроллерами двигателей через интерфейс ввода/вывода общего назначения (General Purpose Input Output, GPIO), с которым мы будем разбираться на протяжении всей *главы 2*. Поскольку Raspberry Pi оснащен разъемом GPIO, имеет небольшие габариты и предназначен для реализации несложных проектов, этот SBC иногда путают с микроконтроллерами типа Arduino. Фактически же это два разных класса устройств.

В чем же разница?

Среди существенных отличий — возможность создавать программное обеспечение, компилировать и выполнять программы непосредственно на Raspberry Pi. В то время как при работе с микроконтроллером Arduino программы создаются и компилируются на внешнем компьютере. Затем они загружаются в микроконтроллер, для чего требуется как минимум USB-кабель, а иногда и специальное устройство — программатор. После этого единственное, что может делать ваш микроконтроллер, — это выполнять данную программу (для которой, кстати, выделен гораздо меньший объем памяти) до тех пор, пока вы не повторите весь процесс. Raspberry Pi способен исполнять несколько программ одновременно. В числе прочих это может быть даже ПО для программирования микроконтроллера. Поскольку все роботы, кроме самых простых, выполняют обычно несколько программ одновременно, мы исключим Arduino и другие микроконтроллеры из списка устройств, подходящих для создания "мозга" нашего робота.

Значит, Raspberry Pi — единственный вариант контроллера для управления роботом?

Вовсе нет. Хотя это превосходный выбор, большая часть программного обеспечения для робота, которое мы рассмотрим в этой книге, будет прекрасно работать на любом другом компьютере без какой-либо модификации. За исключением всего того, что связано с GPIO. Однако на других компьютерах нам приходится использовать внешнее оборудование для выполнения той или иной работы, а это требует другого ПО. Часто бывает экономически невыгодно применять обычный ноутбук и дополнительное оборудование для простого проекта по робототехнике, поскольку возрастают затраты и увеличиваются габариты устройства, но иногда такой компромисс оправдан. Как мы увидим позже, компьютерное зрение требует большой вычислительной мощности. Операции будут выполняться гораздо быстрее, если компьютер оснащен так называемым *графическим процессором* (GPU, Graphic Processor Unit). Эти специализированные процессоры предназначены для выполнения операций с большими блоками видеоданных за один раз, что делает их на порядок быстрее обычного процессора. Компания Nvidia, один из самых известных производителей графических процессоров, в 2019 году выпустила собственный недорогой одноплатный компьютер под названием Jetson Nano. При цене в 100 долларов он оснащается настоящим 128-ядерным графическим процессором и 40-кон-

тактной колодкой GPIO, совместимой с Raspberry Pi, и поэтому неминуемо будет приобретать все большую известность в мире робототехники. Однако звание *идеального обучающего компьютера для робототехники* я все же должен присвоить Raspberry Pi из-за его еще более низкой цены (что делает его доступным для большего числа людей) и мощнейшей поддержки со стороны сетевого сообщества, которой пока нет у Jetson (хотя она быстро растет). Именно поэтому мы выбрали Raspberry Pi в качестве контроллера для наших проектов.

Разве Raspberry Pi не предназначен для школ, энтузиастов-электронщиков и игрушек? Я хотел узнать о настоящей робототехнике.

Вы действительно можете увидеть Raspberry Pi в руках различных любителей мастерить самоделки, но это говорит лишь о его широких возможностях при весьма доступной цене. Университеты любят эти SBC за то, что они позволяют проводить обучение при небольших затратах. Недавно я присутствовал на международном саммите по робототехнике и был удивлен большим числом компаний, связанных с коммерческой робототехникой, которые при разработке автономных роботов выбирают Raspberry Pi в качестве контроллера. На рис. 1.1 показан только один пример.

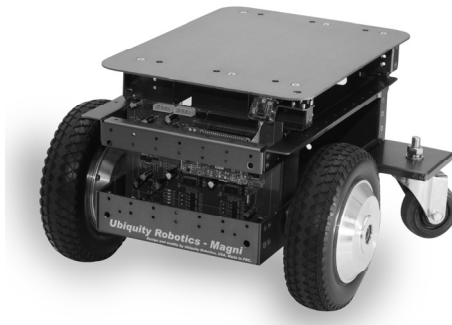


Рис. 1.1. Magni от Ubiquity Robotics — всего лишь один из множества коммерческих роботов на базе Raspberry Pi. Источник: Ubiquity Robotics

Magni и другие коммерческие роботы на базе Raspberry Pi могут выполнять такие функции, как патрулирование, доставка багажа, работа на складе, в офисе, работать в качестве официанта и делать многое другое. Я думаю, вы увидите, что Raspberry Pi гораздо более серьезная вещь, чем просто игрушка, и настоятельно рекомендую начать изучение робототехники именно с него.

Какие модели Raspberry Pi существуют и почему не все из них подходят для наших целей?

Взгляните на примеры плат, изображенные на рис. 1.2. Показаны самые популярные и легкодоступные модели Raspberry Pi. Я не привел здесь менее распространенные (и более старые) модели A и B, которые причислил к категории устройств,

не подходящих для наших целей. Это связано с такими факторами, как низкая скорость, недостаток памяти, а также с тем, что многие пакеты программного обеспечения, которыми мы могли бы воспользоваться, просто не будут на них запускаться. Конечно, их можно было бы заставить работать, но даже в тех случаях, когда они подошли бы, я в целом отдал бы предпочтение таким моделям, как Zero или ZeroW. Это просто более выгодное предложение за те же деньги.

Прежде чем бежать покупать самую мощную модель Raspberry Pi из перечисленных (или новую модель Raspberry Pi 4, которая не представлена на рисунке), дочитайте эту главу до конца. По ряду причин модели 3B+ или 4 могут оказаться не лучшим выбором.





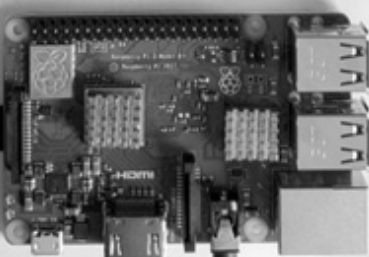
	<p>Raspberry Pi Zero V 1.3 Одноядерный процессор 1 ГГц ОЗУ 512 Мб Один порт микроUSB</p>
	<p>Raspberry Pi Zero W Характеристики, аналогичные Zero 1.3, добавлены встроенные модули Wi-Fi и Bluetooth</p>
	<p>Raspberry Pi 2B Четырехъядерный процессор 900 МГц ОЗУ 1 Гб 4 порта USB Порт Ethernet Отсутствуют встроенные модули Wi-Fi и Bluetooth Благодаря четырехъядерному процессору и большему объему памяти быстродействие модели 2B гораздо выше, чем Zero</p>
	<p>Raspberry Pi 3B Четырехъядерный процессор 1,2 ГГц ОЗУ 1 Гб с большим быстродействием (900 МГц) 4 порта USB Порт Ethernet Встроенные модули Wi-Fi и Bluetooth (Эту модель Raspberry Pi автор считает лучшим выбором для большинства робототехнических проектов)</p>
	<p>Raspberry Pi 3B+ Процессор 1,4 ГГц ОЗУ 1 Гб Также 4 порта USB и порт Ethernet Дополнительный диапазон Wi-Fi 5 ГГц</p>

Рис. 1.2. Сравнение наиболее популярных моделей Raspberry Pi

Raspberry Pi Zero и Raspberry Pi ZeroW

Модели Raspberry Pi Zero и Zero W — замечательные "малыши" за минимальную цену, но, боюсь, у них просто нет тех функций, которые необходимы полноценным контроллерам для роботов, за исключением, возможно, самых простых ботов с обходом препятствий. (Zero W — это обычный Zero со встроенными адаптерами Wi-Fi и Bluetooth). Я обычно использую данные SBC там, где с задачей справился бы, вероятно, и микроконтроллер. Но при этом я получаю возможность удаленно перепрограммировать его или выполнять задачи, для которых он подходит лучше, чем микроконтроллер, например работа с дисплеем или функции доступа в Интернет. Или же в тех случаях, когда требуется длительная автономная работа от батарей — в этом отношении у Zero нет конкурентов среди прочих моделей Pi.

Проблема возникает как из-за недостаточного объема памяти, так и из-за одноядерного процессора. Это не такая большая трудность, если вы профессионал и не страдаете от отсутствия графического интерфейса в Linux. Но если вы привыкли использовать интегрированную среду разработки (Integrated Development Environment, IDE), малышу Zero придется работать на пределе возможностей. Попробуйте запустить вашу IDE через удаленный доступ к рабочему столу (Virtual Network Computing, VNC), и вам придется ждать ответа целую вечность. По этим причинам я использую модели Zero лишь с небольшими программами, которые могу редактировать в обычном текстовом редакторе и компилировать из командной строки.

Кроме того, имейте в виду, что для подключения стандартного HDMI-монитора к моделям Zero вам потребуется переходник. И еще один переходник необходим для подключения стандартного USB-устройства, например, клавиатуры. При работе с менее дорогой моделью Zero 1.3 вам понадобится также отдельный адаптер Wi-Fi, а значит, и USB-концентратор, чтобы одновременно использовать Wi-Fi и клавиатуру. Кроме того, придется купить и распаять на плате коннекторы колодки GPIO. Все это делает модель менее привлекательной, чем кажется на первый взгляд.

Вывод: модель Zero не годится для полностью автономного робота.



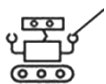
Мы любим модели Raspberry Pi Zero и, бывает, приобретаем их, чтобы удешевить проект. Тем не менее многие новички оказываются разочарованы, когда узнают о необходимости покупки дополнительных адаптеров или видят крайнюю медлительность этих моделей, особенно если используется графическая среда пользователя.

Raspberry Pi 2B

Raspberry Pi 2B — это первая модель, которую я считаю действительно подходящей для создания настоящего автономного робота. Гигабайт памяти и четырехъядерный процессор создают ощущение работы на более-менее нормальном компьютере, а не допотопной технике 1995 года с соединением dial-up и черепашьей скоростью. Вам нужно будет прикупить USB-адаптер для Wi-Fi, если у вас его нет,

и на этом все. В отличие от Zero, плата поставляется с уже распаянными контактами разъема, так что можете сразу же приступить к подключению устройств.

Примечание. Я сказал, что создается ощущение работы на более-менее нормальном компьютере. Однако Pi 2B все же "тормозит" при выполнении некоторых задач, таких как просмотр веб-страниц или удаленный доступ к рабочему столу (VNC).



VNC — это такой способ удаленного управления одним компьютером с помощью другого, причем делать это можно как из соседней комнаты, так и с другой планеты. Существуют и другие способы удаленного управления компьютером, но VNC позволяет совместно использовать графическую среду, так что вы можете работать на удаленной машине так, словно подключены непосредственно к ней. Это может быть очень удобно при разработке роботов.

Вывод: Raspberry Pi 2B — это приемлемый выбор. Но если у вас пока нет USB-адаптера Wi-Fi, можно приобрести еще более быструю модель Raspberry Pi 3B примерно за ту же цену. Даже если для вашего робота не требуется высокопроизводительный компьютер, вы оцените преимущество более быстрого процессора, будь то разработка и тестирование кода или запуск готового проекта (можно подумать, эти проекты когда-нибудь действительно будут закончены). Я терпеть не могу, когда после нажатия на кнопку приходится бесконечно ждать результата. Начинаю отвлекаться на что-то другое и в конце концов оказываюсь на улице, гоняясь со своей собакой за белками, так и не сделав ничего путного.

Raspberry Pi 3B — лучший выбор!

На мой взгляд, Raspberry Pi 3B — лучший выбор, т. к. обеспечивает оптимальное сочетание энергопотребления, скорости и встроенных функций. Кстати, это первая модель с установленными модулями Wi-Fi и Bluetooth, что избавляет вас от необходимости покупать USB-адаптеры. Кажется, что новый, более быстрый процессор работает прямо молниеносно по сравнению с моделью 2B. В то время как модель 2B может обходиться только радиатором, для охлаждения платы 3B лучше предусмотреть радиатор и небольшой вентилятор, особенно при размещении в корпусе. Если чипы перегреваются, они автоматически переходят на более низкие частоты, так что в принципе дополнительное охлаждение не требуется, но радиаторы и вентиляторы можно приобрести всего за пару долларов. Эти небольшие затраты окупаются за счет более высокой производительности.

Вывод: Raspberry Pi 3B — это победитель! Из всех моделей Raspberry Pi я выбираю именно модель 3B для всех своих проектов мобильных роботов.

Raspberry Pi 3B+

Несмотря на более быстрый процессор, Raspberry Pi 3B+ — не лучший выбор в качестве контроллера для управления роботом, особенно мобильным. Это прекрасная машина, но результаты множества неофициальных тестов показывают, что незначительный (около 10–15%) прирост производительности достигается за счет роста

потребления энергии на 50%. Это существенный недостаток для мобильного робота, работающего от аккумуляторных батарей. Хотя я не замерял по часам, за какое время разряжается аккумулятор, разница заметна. Дополнительное энергопотребление означает также, что вам придется улучшить охлаждение. Конечно, приятно, что в Raspberry Pi 3B+ имеется 2-й диапазон Wi-Fi, но эта функция не обязательна.

Вывод: все намеченное можно прекрасно реализовать как на модели Raspberry Pi 3B+, так и на 3B (или 2B, если уж на то пошло), так что выбор за вами. Я держу свою модель 3B+ на столе, где она может оставаться подключенной к розетке, и использую обычную Raspberry Pi 3B для своих роботов.

Новая модель Raspberry Pi 4

Вскоре после того, как я начал писать эту книгу, компания Raspberry Pi Foundation выпустила совершенно новую модель 4B. Более быстрый процессор, увеличение объема памяти до 4 Гб и другие значительные аппаратные изменения делают эту машину значительно более производительной. Но потребовалось существенно пересмотреть программные пакеты для обеспечения их совместимости. В настоящее время (февраль 2020 года, примерно семь месяцев спустя) разработчики операционной системы и другого программного обеспечения все еще пытаются добиться совместимости пакетов и операционных систем, которые мы используем в робототехнике. Можно заставить их работать, но забудьте о легких методах установки — даже для простого робота мне пришлось выполнить компиляцию и установку множества стандартных пакетов из исходников.

Даже если ПО догонит аппаратное обеспечение и его использование станет таким же простым, как в модели Pi 3B, я не вижу большого преимущества из-за повышенного энергопотребления. А оно, по общему мнению, крайне велико (даже по сравнению с моделью 3B+).

Хотя у модели Pi 4B отличный потенциал, она просто еще не готова для тех из нас, кого больше волнуют реальные проекты, а не "танцы с бубном" при установке программного обеспечения. Я искренне не советую применять Pi 4B тем, кто на данный момент только начинает осваивать Linux. Если вам нужна более мощная машина, рекомендую обратить внимание на Nvidia Jetson Nano — эта модель гораздо более мощная, значительно лучше поддерживается, а стоит лишь немного дороже. Единственное, для чего я использовал бы Pi 4 в работе, — для ускорения функций компьютерного зрения и машинного обучения, а Jetson Nano как раз для этого и предназначен.

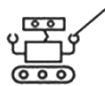
Выбор операционной системы

Для Raspberry Pi, как и для любого другого компьютера, можно выбирать различные операционные системы. Боюсь, что Windows в их число не входит — она слишком тяжеловесна для этой маленькой машины. Некоторые экспериментируют с установкой на Raspberry Pi определенных версий Android, но, как я понимаю, при

этом часто возникают ошибки и сбои. Давайте поговорим о двух версиях операционной системы Linux.

Raspbian

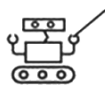
Если вы хоть немного работали с Raspberry Pi, то почти наверняка знакомы с официальной ОС для нее — Raspbian. Это, безусловно, самая популярная операционная система для Raspberry Pi, имеющая самую большую поддержку со стороны сообщества. Raspbian — отличная отправная точка для знакомства с Linux и Raspberry Pi, и я рекомендую вам уделить некоторое время небольшим обучающим руководствам, которые покажутся вам интересными, чтобы освоиться с Raspbian, Linux в целом и с самой Raspberry Pi. Однако я не рекомендую эту операционную систему для робототехники и не собираюсь использовать ее в наших проектах в этой книге. Скоро вы поймете почему.



Raspbian — это название линейки пользовательских дистрибутивов Linux от компании Raspberry Pi Foundation (называемых для краткости *distro*). Было создано множество их версий или релизов, поэтому, обращаясь в Интернет за помощью в устранении неполадок, необходимо указать, с каким именно релизом возникла проблема. Релизам присваиваются имена, например *Jessie* или из недавних (по состоянию на февраль 2020 года) — *Buster*. Дистрибутивы Raspbian основаны на версиях Linux Debian.

Raspbian определенно подходит для больших робототехнических проектов, которые мы собираемся реализовать. Тем не менее это потребует либо гораздо большего объема программного кода, либо умения настраивать и подправлять устанавливаемые программы. Я же просто хочу запускать своих роботов, поэтому не буду лезть в дебри, но если вы получаете удовольствие от процесса, заставляя эти штуки работать, дерзайте!

Минималисты могут также выбрать *lite*-версию различных вариантов Raspbian. Это "голая" операционная система без каких-либо автоматически устанавливаемых дополнительных программных пакетов и даже без графического интерфейса — все выполняется через окно терминала.



Большим преимуществом Raspberry Pi, работающей с картой microSD, является то, что карты легко менять. Я часто путешествую с Raspberry Pi, и быстрая замена SD-карты равносильна получению совершенно другой машины для других целей.

Ubuntu

Новички в Linux часто удивляются обилию версий этой ОС. Debian, Ubuntu, Mint, Red Hat, Fedora, CentOS — это лишь малая часть. Кроме того, каждая из этих операционных систем также имеет множество версий. Мы называем их "ароматы" (*flavors*).

Существует множество ароматов Ubuntu, я хочу остановиться на двух из них: полная версия Ubuntu Desktop и более легковесная версия, называемая *Lubuntu*.