

Оглавление

Благодарности	9
Отзывы и обратная связь	10
Глава 1. Введение.....	11
Raspberry Pi Pico.....	12
Источник примеров кода	14
Использование примеров	14
О чем рассказывает эта книга	15
Глава 2. Использование датчиков	16
Измеряем то, что нам действительно нужно получить	16
Датчик может влиять на то, что вы измеряете.....	16
Погрешность и разрешающая способность.....	17
Ошибки АЦП	18
Усреднение показаний.....	18
Калибровка.....	20
Проверка показаний	20
Глава 3. Программирование Raspberry Pi Pico.....	22
Библиотеки.....	22
Официальный пакет C/C++ SDK.....	23
Arduino IDE.....	24
MicroPython.....	28
CircuitPython	29
Thonny IDE	30
Настройка Thonny	31
Доступ к REPL	31
Ввод, редактирование и сохранение программ	31
Запуск программы.....	31
Перемещение файлов между Pico и ПК	32
Просмотр выходных данных программ.....	32
Пакет C/C++ SDK	32
Arduino	33
MicroPython и CircuitPython	33
Какую среду следует использовать?	33
Глава 4. Интерфейсы и протоколы.....	34
Немного об электронике	34
Общие разновидности интерфейсов	36

АЦП – аналого-цифровой преобразователь	37
Внутренний датчик температуры	37
Использование АЦП с пакетом C/C ++ SDK.....	38
Использование АЦП в среде Arduino	39
Использование АЦП в MicroPython	40
Использование АЦП в CircuitPython	40
Подключение цифровых сигналов от датчиков к Pico.....	40
Цифровой вход и выход общего назначения (GPIO)	41
Использование GPIO в пакете C/C++ SDK	42
Использование GPIO в среде Arduino	42
Использование GPIO в MicroPython.....	43
Использование GPIO в CircuitPython	43
UART – универсальный асинхронный приемопередатчик	44
Формат кадра UART	44
Управляющие сигналы и аппаратное управление потоком данных.....	45
Назначение выводов	46
Использование UART в пакете C/C++ SDK.....	46
Использование UART в Arduino IDE	47
Использование UART в MicroPython	48
Использование UART в CircuitPython.....	49
Последовательный периферийный интерфейс SPI	50
Сигналы SPI.....	51
Режимы SPI	52
SPI в Raspberry Pi Pico.....	53
Параметры выводов	53
Использование SPI с пакетом C/C++ SDK.....	53
Использование SPI в Arduino IDE	55
Использование SPI в MicroPython	55
Использование SPI в CircuitPython	57
Последовательный интерфейс I ² C	58
Топология I ² C	58
Физический интерфейс I ² C	59
Операции чтения и записи в I ² C.....	60
I ² C в Raspberry Pi Pico	63
Использование I ² C с пакетом C/C++ SDK.....	63
Использование I ² C в Arduino IDE	64
Использование I ² C в MicroPython.....	65
Использование I ² C в CircuitPython	66
Поиск I ² C-адреса устройства.....	68
Глава 5. Датчики с релейным выходом	69
Прочтите это, прежде чем приступать к примерам.....	69
Кнопки.....	70
Пример использования кнопки.....	71
Пример отслеживания кнопки (C/C++ SDK)	72
Пример отслеживания кнопки (Arduino IDE)	74
Пример отслеживания кнопки (MicroPython)	75
Пример отслеживания кнопки (CircuitPython)	76

Герконовый выключатель	77
Клавиатуры	77
Пример чтения клавиатуры.....	80
Пример чтения клавиатуры (C/C++ SDK)	81
Пример чтения клавиатуры (Arduino IDE)	84
Пример чтения клавиатуры (MicroPython)	86
Пример чтения клавиатуры (CircuitPython)	88
Датчик вибрации SW-420.....	89
Пример датчика вибрации	90
Пример датчика вибрации (C/C++ SDK)	90
Пример датчика вибрации (Arduino IDE)	91
Пример датчика вибрации (CircuitPython).....	93
Датчик присутствия PIR.....	93
Пример подключения датчика присутствия (PIR)	94
Пример PIR-датчика (C/C++ SDK)	95
Пример PIR-датчика (Arduino IDE)	96
Пример PIR-датчика (MicroPython)	96
Пример PIR-датчика (CircuitPython)	97
Датчик пламени.....	97
Пример использования датчика пламени	98
Пример датчика пламени (C/C++ SDK)	98
Пример датчика пламени (Arduino IDE)	99
Пример датчика пламени (MicroPython)	100
Пример датчика пламени (CircuitPython)	100
Цифровой датчик звука	100
Пример подключения цифрового датчика звука.....	101
Пример датчика звука (C/C++ SDK)	101
Пример датчика звука (Arduino IDE)	102
Пример датчика звука (MicroPython)	103
Пример датчика звука (CircuitPython)	103
Датчики газа MQ.....	103
Пример подключения датчика газа MQ	105
Пример датчика газа MQ (C/C++ SDK)	106
Пример датчика газа MQ (Arduino IDE)	106
Пример датчика газа MQ (MicroPython)	107
Пример датчика газа MQ (CircuitPython).....	107
Цифровые датчики Холла	108
Модуль KY-003	108
Датчик Холла US1881	109
Пример использования датчика на основе эффекта Холла	109
Пример датчика Холла (C/C++ SDK)	111
Пример датчика Холла (Arduino IDE)	114
Пример датчика Холла (MicroPython)	116
Пример датчика Холла (CircuitPython)	118
Глава 6. Аналоговые датчики	121
Потенциометры	121

Пример подключения потенциометра	122
Пример подключения потенциометра (C/C++ SDK)	123
Пример подключения потенциометра (Arduino IDE)	125
Пример подключения потенциометра (MicroPython)	125
Пример подключения потенциометра (CircuitPython).....	126
Аналоговые джойстики	126
Пример подключения аналогового джойстика	127
Пример аналогового джойстика (C/C++ SDK)	127
Пример аналогового джойстика (Arduino IDE)	130
Пример аналогового джойстика (MicroPython)	131
Пример аналогового джойстика (CircuitPython).....	132
Фоторезисторы (LDR)	133
Пример подключения фоторезистора	134
Пример фоторезистора (C/C++ SDK)	134
Пример фоторезистора (Arduino IDE)	135
Пример фоторезистозпа (MicroPython)	136
Пример фоторезистора (CircuitPython)	137
Фототранзистор	137
Пример подключения фототранзистора	138
Пример фототранзистора (C/C++ SDK)	139
Пример фототранзистора (Arduino IDE)	140
Пример фототранзистора (MicroPython и CircuitPython).....	140
Использование светодиодов в качестве	
датчиков освещенности	140
Пример светодиода в качестве датчика освещенности.....	141
Пример светодиода как датчика освещенности (C/C++ SDK)	142
Пример светодиода как датчика освещенности (Arduino IDE)	142
Пример светодиода как датчика освещенности (MicroPython)	143
Пример светодиода как датчика освещенности (CircuitPython).....	143
Датчик газа.....	144
Пример аналогового датчика газа.....	144
Аналоговый датчик Холла	145
Пример подключения аналогового датчика Холла.....	145
Пример аналогового датчика Холла (C/C++ SDK)	146
Пример аналогового датчика Холла (Arduino IDE)	147
Пример аналогового датчика Холла (MicroPython)	148
Пример аналогового датчика Холла (CircuitPython).....	148
Глава 7. Датчики температуры	150
Термистор	150
Пример подключения термистора	152
Пример термистора (C/C++ SDK)	153
Пример термистора (Arduino IDE)	155
Пример термистора (MicroPython)	156
Пример термистора (CircuitPython)	157
Датчики температуры LM35D и TMP36	158

Пример подключения LM35D и TMP36	159
Пример LM35D и TMP36 (C/C++ SDK)	160
Пример LM35D и TMP36 (Arduino IDE)	161
Пример LM35D и TMP36 (MicroPython)	162
Пример LM35D и TMP36 (CircuitPython)	163
Датчик температуры DS18B20	163
Протокол 1-Wire	165
Команды доступа к ROM	165
Функциональные команды DS18B20	166
Пример подключения DS18B20	169
Пример DS18B20 (C/C++ SDK)	170
Пример DS18B20 (Arduino IDE)	171
Пример DS18B20 (MicroPython)	174
Пример DS18B20 (CircuitPython)	174
Датчики температуры и влажности DHT11 и DHT22	175
Пример подключения DHT11 и DHT22	177
Пример DHT11 и DHT22 (C/C++ SDK)	178
Пример DHT11 и DHT22 (Arduino IDE)	182
Пример DHT11 и DHT22 (MicroPython)	184
Пример DHT11 и DHT22 (CircuitPython)	187
Датчик температуры LM75A	188
Пример подключения LM75A	190
Пример LM75A (C/C++ SDK)	191
Пример LM75A (Arduino IDE)	193
Пример LM75A (MicroPython)	194
Пример LM75A (CircuitPython)	195
Датчик температуры и влажности HDC1080	197
Пример подключения HDC1080	199
Пример HDC1080 (C/C++ SDK)	200
Пример HDC1080 (Arduino IDE)	202
Пример HDC1080 (MicroPython)	203
Пример HDC1080 (CircuitPython)	204
Датчик температуры MCP9808	205
Функции сигнализации Alert	208
Пример подключения MCP9808	209
Пример датчика MCP9808 (C/C++ SDK)	210
Пример датчика MCP9808 (Arduino IDE)	212
Пример датчика MCP9808 (MicroPython)	214
Пример датчика MCP9808 (CircuitPython)	215
Датчик температуры и влажности AHT10	216
Пример подключения AHT10	218
Пример AHT10 (C/C++ SDK)	218
Пример AHT10 (Arduino IDE)	220
Пример AHT10 (MicroPython)	221
Пример AHT10 (CircuitPython)	222
Сравнение датчиков	223

Глава 8. Датчики атмосферного давления	224
BMP085 и BMP180	226
Пример подключения BMP085 и BMP180	228
Пример датчиков BMP085/BMP180 (C/C++ SDK)	228
Пример датчиков BMP085/BMP180 (Arduino IDE)	234
Пример датчиков BMP085/BMP180 (MicroPython)	235
Пример датчиков BMP085/BMP180 (CircuitPython).....	236
BMP280 и BME280	236
Примеры подключения BMP280 и BME280.....	241
Пример BMP280/BME280 (C/C++ SDK)	241
Пример BMP280/BME280 (Arduino IDE)	251
Пример BMP280/BME280 (MicroPython)	252
Пример BMP280/BME280 (CircuitPython)	253
Датчик BMP390	253
Пример подключения BMP390	258
Пример BMP390 (C/C++ SDK)	258
Пример BMP390 (Arduino IDE)	262
Пример BMP390 (MicroPython)	263
Пример BMP390 (CircuitPython)	264
Глава 9. Электронные компасы, акселерометры и гироскопы	266
3-осевые магнитные датчики HMC5883L, HMC5983 и QMC5883L	267
Регистры HMC5883L и HMC5983 и операции с ними.....	270
Регистры QMC5883L и операции с ними	273
Пример подключения HMC5883L, HMC5983 и QMC5883L	274
Пример HMC5883L, HMC5983 и QMC5883L (C/C++ SDK)	275
Пример HMC5883L и QMC5883L (Arduino IDE)	280
Пример QMC5883L (MicroPython)	281
Пример QMC5883L (CircuitPython)	282
3-осевой акселерометр ADXL345	283
Пример подключения ADXL345	286
Пример ADXL345 (C/C++ SDK)	286
Пример ADXL345 (Arduino IDE)	290
Пример ADXL345 (MicroPython)	291
Пример ADXL345 (CircuitPython).....	292
3-осевой акселерометр MMA8452	293
Пример подключения MMA8452	296
Пример MMA8452 (C/C++ SDK)	297
Пример MMA8452 (Arduino IDE)	300
Пример MMA8452 (MicroPython)	301
Пример MMA8452 (CircuitPython)	306
3-осевой акселерометр и гироскоп MPU6050	312
Пример подключения MPU6050	314
Пример MPU6050 (C/C++ SDK)	315
Пример MPU6050 (Arduino IDE)	319

Пример MPU6050 (MicroPython)	320
Пример MPU6050 (CircuitPython)	322
Сравнение электронных компасов, акселерометров и гироскопов	323
Глава 10. Различные датчики.....	324
Ультразвуковой датчик HC-SR04	324
Датчик HY-SRF05	326
Пример подключения HC-SR04	326
Пример HC-SR04 (C/C++ SDK)	327
Пример HC-SR04 (C/C++ SDK): программирование PIO	327
Пример HC-SR04 (C/C++ SDK): основная программа	329
Пример HC-SR04 (Arduino IDE)	330
Пример HC-SR04 (MicroPython)	331
Пример HC-SR04 (CircuitPython)	332
Датчик угла поворота (энкодер)	333
Пример подключения датчика угла поворота.....	335
Пример датчика угла поворота (C/C++ SDK)	336
Пример датчика угла поворота (C/C++ SDK): программирование PIO	336
Пример датчика угла поворота (C/C++ SDK): основная программа	338
Пример датчика угла поворота (Arduino IDE)	340
Пример датчика угла поворота (MicroPython)	342
Пример датчика угла поворота (CircuitPython).....	344
Датчик нагрузки (тензодатчик)	345
АЦП HX711	346
Пример подключения тензодатчика.....	348
Пример тензодатчика (C/C++ SDK)	349
Пример тензодатчика (Arduino IDE)	352
Пример тензодатчика (MicroPython)	354
Пример тензодатчика (CircuitPython).....	356
Ключи iButton	358
Пример подключения iButton	358
Пример iButton (C/C++ SDK)	359
Пример iButton (Arduino IDE)	360
Пример iButton (MicroPython)	362
Пример iButton (CircuitPython)	363
Глава 11. Датчики отпечатков пальцев.....	364
Основы идентификации по отпечаткам пальцев	364
Протокол датчика FPM10A.....	366
Конфигурация датчика	367
Регистрация отпечатка пальца.....	368
Идентификация отпечатка пальца	369
Передача образа и шаблона	369
Пример подключения датчика отпечатков пальцев.....	370
Пример датчика отпечатков пальцев (C/C++ SDK)	371

Пример датчика отпечатков пальцев (Arduino IDE)	375
Пример датчика отпечатков пальцев (MicroPython)	378
Пример датчика отпечатков пальцев (CircuitPython).....	382
Глава 12. Радиочастотные идентификаторы RFID	384
Система RFID 125kHz.....	384
Пример подключения RFID 125 кГц	386
Пример RFID 125 кГц (MicroPython)	387
Пример RFID 125 кГц (SDK C/C++)	390
Пример RFID 125 кГц (Arduino IDE)	393
Пример RFID 125 кГц (CircuitPython)	397
Системы MIFARE/NFC.....	400
Адресация памяти.....	401
Конфигурационный блок.....	401
Оформление билетов	403
Считыватель на основе MFRC522	405
Считыватель PN532	405
Примеры подключения MIFARE	406
Чтение и запись данных в MicroPython с помощью считывателя MFRC522	406
Пример считывателя RFID-RC422 (MicroPython)	408
Оформление билетов с помощью считывателя MFRC522 в среде Arduino IDE.....	409
Считывание и запись данных в CircuitPython с помощью считывателя PN532	416
Глава 13. Итоги	420
Правильный выбор датчиков и их использование	420
Использование нового датчика	420
Написание кода и использование библиотек.....	421
Что дальше?.....	421
Приложение А. Другие платы на базе микроконтроллера RP2040	423
Adafruit Feather RP2040	423
SeeedStudio XIAO RP2040	424
Arduino Nano RP2040 Connect	425
Raspberry Pi Pico W	426
Приложение В. Неоригинальные датчики.....	428
Неоригинальные датчики температуры DS18B20.....	428
Поддельный акселерометр MPU6050	429
Предметный указатель	430

Благодарности

Оглядываясь назад, я понимаю, что было слишком много людей, которые так или иначе помогли мне прийти к возможности написать эту книгу. Здесь я упоминаю некоторых из них и благодарю их.

Во-первых, мои мать и отец развили во мне любознательность и пристрастили к чтению.

Было много учителей, которые не только давали важные уроки, но и побуждали меня узнавать больше.

В своей профессиональной жизни я благодарен всем, кто верил, что я могу что-то сделать, и тем, кто помогал мне в этом.

Хочу особо отметить покойного Альберто Фабиано (Alberto Fabiano), который познакомил меня с бурным сообществом хакеров. Фабио Соуса (Fabio Souza) и Тьяго Лима (Tiago Lima) с бразильского портала Embarcados оказали мне большую поддержку в написании статей.

Также большим вдохновителем оказался мой зять Маурисио Анише (Mauricio Aniche).

Конечно, эта книга не появилась бы на свет, если бы не терпение моей жены Сесилии, в то время как я целыми днями сидел за компьютером и играл со всеми этими макетками.

Отзывы и обратная связь

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

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

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в тексте, пожалуйста, сообщите о них по адресу dmkpress@gmail.com, и мы исправим это в следующих тиражах.

Глава 1

Введение

Датчики позволяют встраиваемой системе получать представление о том, что происходит в физическом мире. Обычно данные с датчиков обрабатываются микроконтроллером и используются для принятия решения о том, какие действия следует предпринять.

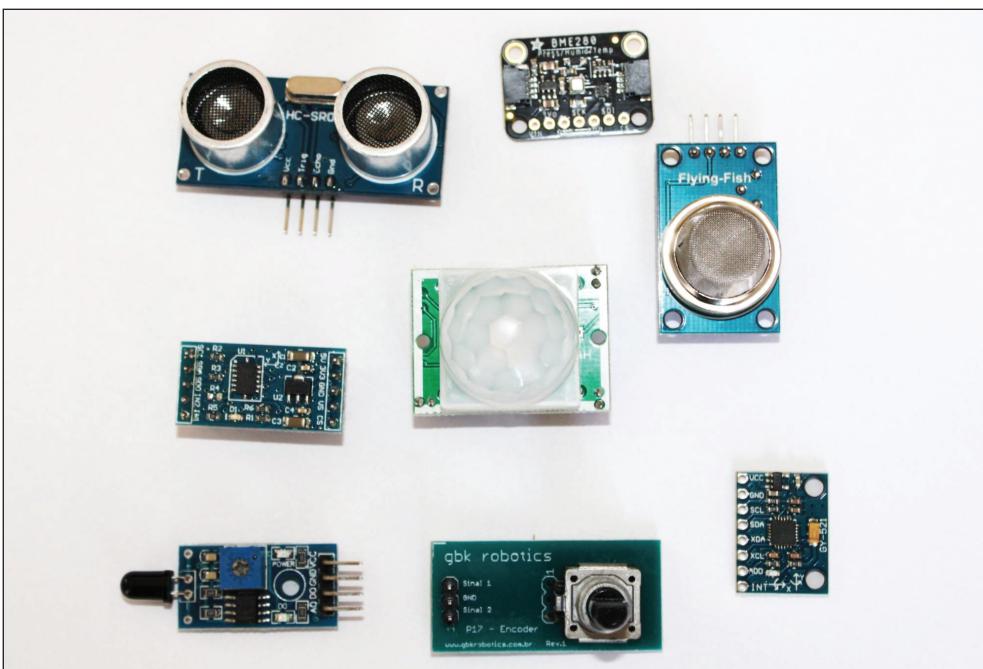


Рис. 1.1. Некоторые датчики

Из этой книги вы узнаете о датчиках в общем и о множестве способов, которыми они могут передавать данные на микроконтроллер. Вы также познакомитесь с некоторыми типичными стратегиями своевременного получения надежных данных. Более подробно вы узнаете о ресурсах, доступных на плате Raspberry Pi Pico для подключения датчиков, и о том, как их использовать при программировании на C/C++, MicroPython и CircuitPython. Затем мы рассмотрим несколько популярных датчиков более подробно.

Я постарался охватить как можно большее количество наиболее распространенных датчиков. Конечно, каждый день на рынке появляются новые, а найти некоторые старые датчики становится все труднее. Это одна из причин, по которой я подробно разъясняю работу с датчиками: благодаря этим знаниям у вас будет меньше трудностей при использовании новой разработки.

Я расскажу о том, как использовать интерфейсы Pico в коде, и вы увидите много примеров кода, но изучение языков программирования не является целью этой книги. Предполагается, что вы обладаете базовыми знаниями C или Python, это позволит лучше понимать примеры кода и писать на их основе собственные программы.

RASPBERRY PI PICO

Raspberry Pi Pico был выбран в качестве базовой платы контроллера для этой книги по многим причинам: это легкодоступная, дешевая и мощная плата. Под мощностью я подразумеваю не только то, что Pico имеет большое количество интерфейсов для подключения датчиков и серьезную вычислительную мощность, но и то, что эта вычислительная мощность и объем доступной памяти позволяют нам использовать такие сложные языки, как MicroPython и CircuitPython.

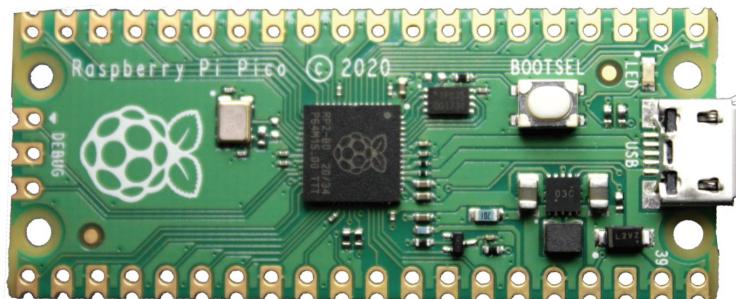


Рис. 1.2. Raspberry Pi Pico

В Pico используется микроконтроллер RP2040. Он оснащен двумя ядрами ARM Cortex M0+, 264 Кбайтами оперативной памяти и большим количеством периферийных устройств (GPIO, таймеры, PWM, UART, I²C, SPI, АЦП и USB)¹.

Одной из важных особенностей микроконтроллера RP2040 является блок PIO (*Programmable In/Out*, программируемый ввод-вывод), который позволяет реализовать простые аппаратные протоколы в виде коротких программных процедур, работающих параллельно основным ядрам ARM. В этой книге я не буду вдаваться в подробности программирования PIO. Тем, кто заинтересуется, рекомендую мою книгу «Знакомство с RP2040»².

¹ Разъяснение приведенных сокращений см. в главе 4 «Интерфейсы и протоколы». — Здесь и далее примечания переводчика.

² Daniel Quadros «Knowing the RP2040. A Guide for Programmers», <https://leanpub.com/rp2040>.

Хотя рисунки и текст относятся к модели Raspberry Pi Pico, большая часть представленной информации и методов может быть использована с другими платами, использующими микроконтроллер RP2040. Ознакомиться с некоторыми из них можно в [приложении А](#).

C/C++ давно зарекомендовал себя как язык низкого уровня, который лучше всего подходит для встраиваемых систем, обеспечивая производительность и предсказуемость (за счет написания большего количества строк кода). MicroPython обеспечивает микроконтроллерам простоту использования и мощь языка Python. Оба они получают прямую поддержку от разработчиков Pico. CircuitPython – это форк (ответвление) MicroPython. Хотя фирма Raspberry Foundation напрямую не поддерживает Python для Pico, предлагаются модули ввода-вывода, которые упрощают перенос кода между различными платами, а также имеется огромное количество драйверов и примеров кода от Adafruit.

На рис. 1.3 (взятом из официальной документации) схематично показан Raspberry Pi Pico и сигналы на его выводах. Можно заметить, что некоторые функции (например, UART0 TX) отображаются на нескольких выводах (в данном случае на 1, 16 и 21). В рамках инициализации некоторых периферийных устройств мы выберем тот вывод, который использовать наиболее удобно.

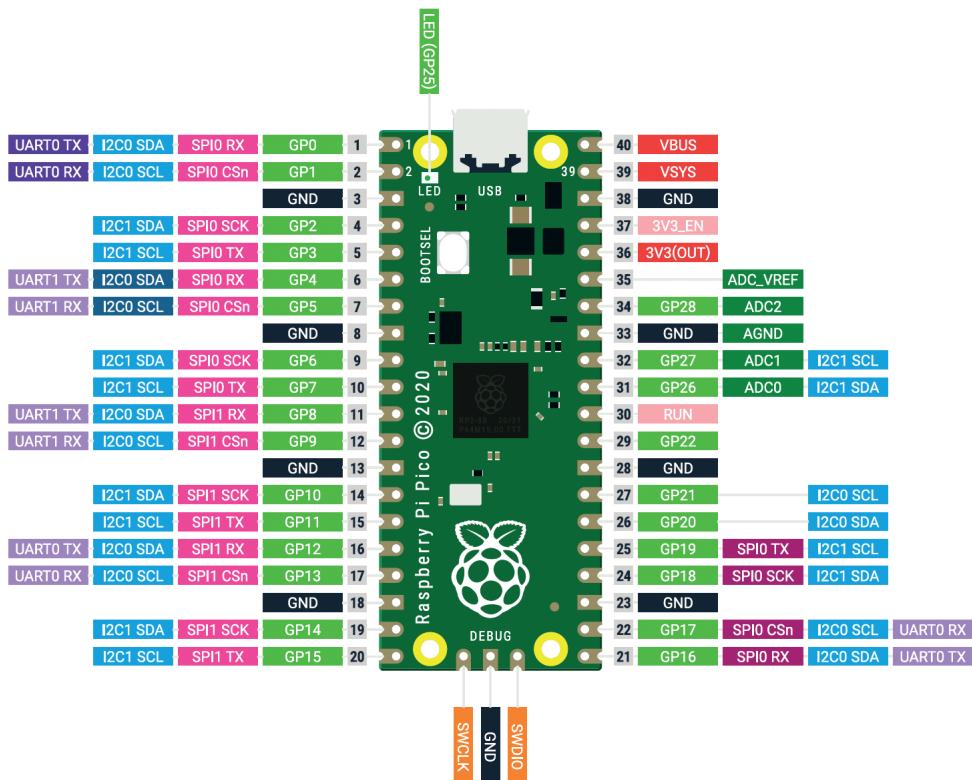


Рис. 1.3. Разводка выводов Raspberry Pi Pico

Если вы включаете (или сбрасываете) Pico с помощью кнопки BOOTSEL, она переводит контроллер в специальный режим.

В примерах из этой книги мы собираемся подключить Pico к ПК через USB. Это подключение будет использоваться для:

- питания платы (и подключенных к ней датчиков);
- загрузки программ на плату;
- взаимодействия с программой, запущенной на Pico.

Мы подробнее поговорим обо всем этом в главе 3 «Программирование Raspberry Pi Pico».

Источник примеров кода

В этой книге я привожу только наиболее важные части кода. Полный текст кода примеров можно загрузить по адресу: <https://github.com/dquadros/PicoSensors>.

Использование примеров

Самый простой способ проверить примеры – собрать схемы на макетной плате.

Макетная плата – это пластиковая плата с отверстиями, в которые можно вставлять выводы компонентов, и внутренними соединениями между этими отверстиями. Провода (перемычки) используются для выполнения соединений, которые невозможно выполнить с помощью внутренних соединений.

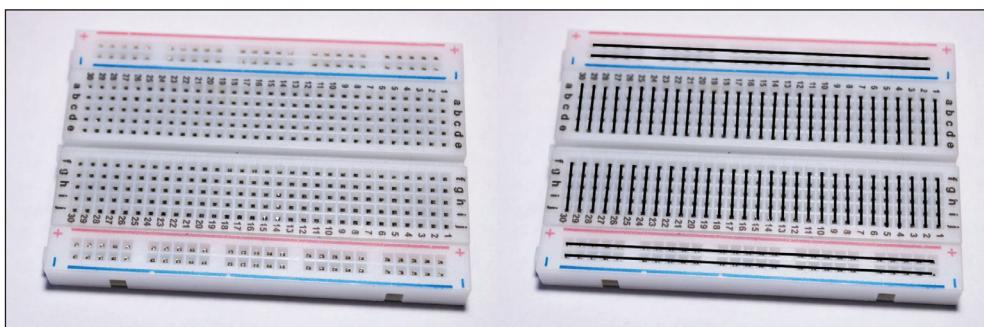


Рис. 1.4. Макетная плата

Как только вы поймете, какие отверстия имеют внутренние соединения, вам будет легко изменять положение компонентов¹.

Чтобы использовать Pico и датчики на макетной плате, к их контактам должны быть припаяны штыревые разъемы. В некоторых случаях, вместо того чтобы вставлять датчик непосредственно в макетную плату, лучше использовать провода-перемычки, тогда или в платах датчиков используются разъемы гнездового типа, или подбираются перемычки с гнездовыми разъемами.

Ко всем примерам приложены наглядные монтажные схемы (сделанные с помощью программы Fritzing²), показывающие соединения на макетной плате.

¹ На рис. 1.4 справа внутренние соединения показаны черными линиями.

² См. <https://fritzing.org/>.

СОВЕТ. На Pico нет кнопки сброса, но вы можете подключить ее между выводом 30 (RUN) и GND. Некоторые маленькие кнопки имеют точное расстояние между выводами, совпадающее с расстоянием между контактами 30 и 28, и вы можете использовать одну из них в макете или припаять непосредственно к плате¹.

О ЧЕМ РАССКАЗЫВАЕТ ЭТА КНИГА

Это «*Введение*» (глава 1) дает общее представление о том, что такое датчики и что вы найдете в данной книге.

В главе 2 «*Использование датчиков*» в общих чертах рассказывается о том, почему датчики иногда не дают нужных результатов и как их можно улучшить.

В главе 3 «*Программирование Raspberry Pi Pico*» рассказывается о том, как мы можем писать программы для Pico. В этой книге мы будем использовать четыре варианта: официальный C/C++ SDK, Arduino IDE, MicroPython и CircuitPython.

В главе 4 «*Интерфейсы и протоколы*» мы рассмотрим различные способы передачи данных на микроконтроллер и способы доступа к ним в указанных четырех средах программирования.

В следующих главах речь пойдет о конкретных датчиках. Можно организовать датчики различными способами, основываясь на способах их подключения или параметрах, которые они измеряют. Я использовал смешанный способ.

В главе 5 «*Датчики с релейным выходом*» описаны датчики, которые имеют один выход, выдающий сигнал «да»/«нет» или «включено»/«выключено». Хорошим примером может служить кнопка, которая подает сигнал, если ее нажать или отпустить.

В главе 6 «*Аналоговые датчики*» рассказывается о датчиках, которые выдают аналоговый сигнал, например о датчиках освещенности, которые сообщают, насколько светло или темно на улице.

В главе 7 «*Датчики температуры*» мы перейдем к классификации по измеряемому параметру и для начала поговорим о датчиках температуры.

Глава 8 «*Датчики атмосферного давления*» посвящена барометрическим датчикам, а глава 9 «*Электронные компасы, акселерометры и гироскопы*» – акселерометрам и компасам.

В главе 10 «*Различные датчики*» рассказывается о некоторых датчиках, которые не были представлены в предыдущих главах.

Глава 11 «*Датчики отпечатков пальцев*» посвящена датчикам отпечатков пальцев, а в главе 12 «*RFID*» вы узнаете о двух типах бесконтактных RFID-датчиков.

Глава 13 «*Итоги*» завершает повествование кратким обзором пройденного и некоторыми напоминаниями об использовании датчиков в проекте.

В приложении А «*Некоторые другие платы на базе RP2040*» показано несколько альтернативных плат на базе микропроцессора RP2040.

В приложении В «*Неоригинальные датчики*» поднимается проблема, с которой, я надеюсь, вам не придется иметь дело: о клонах и подделках оригинальных датчиков. Здесь рассказывается о двух неудачных примерах, с которыми пришлось столкнуться при тестировании примеров для этой книги.

¹ Подойдут некоторые конструкции т. н. «тактовых» кнопок (см., например, <https://www.chipdip.ru/product/kls7-ts6607-4.3-180-t-it-1102-rt>).

Глава 2

Использование датчиков

Много лет назад я получил новый магнитный датчик, который должен был определять положение относительно северного магнитного полюса, подобно компасу. Я изучил документацию и написал код, чтобы получить показания. Датчик выдал число, и, снова следя документации, я преобразовал это число в угол направления. Я провел несколько ориентировочных тестов, и результаты измерений изменились, когда я повернул датчик. Все в порядке, не так ли?

Некоторое время спустя с помощью этого датчика я решил сконструировать компас, показывающий направление на север с помощью кольца из 12 светодиодов. К моему разочарованию, я не смог заставить его работать точно. Показания сильно «гуляли», когда датчик находился в фиксированном положении, линейной зависимости между показаниями не было, а поворот и близлежащие объекты могли изменить результаты!

Для получения хороших результатов от датчиков требуется гораздо больше, чем просто получить от них данные. В этой главе я расскажу о некоторых трудностях на этом пути и способах их преодоления.

ИЗМЕРЯЕМ ТО, ЧТО НАМ ДЕЙСТВИТЕЛЬНО НУЖНО ПОЛУЧИТЬ

Внутри RP2040 находится датчик температуры. С помощью нескольких строк кода мы можем получить показания. Но что мы измеряем? Датчик находится внутри микросхемы, поэтому он даст лучшее представление о внутренней температуре, чем о температуре окружающей среды. Действительно, в большинстве случаев температура будет значительно превышать температуру окружающей среды. Показания будут увеличиваться, когда будут работать ядра микропроцессора ARM.

Расположение датчика имеет основополагающее значение для измерения того, что вы хотите получить. Во многих случаях свойство, которое вы хотите измерить, зависит от местоположения. Вам решать, какое расположение является важным для данного приложения. Может потребоваться некоторое тестирование, чтобы узнать, как меняется измеряемая величина и какое расположение является наилучшим. В некоторых случаях вам может потребоваться использовать несколько датчиков и среднее значение или более сложную логику, применяющую все значения.

Датчик может влиять на то, что вы измеряете

Хотя этот эффект встречается не так часто, вам следует обратить внимание на случаи, когда датчик может влиять на измеряемую величину. Одним из приме-

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru