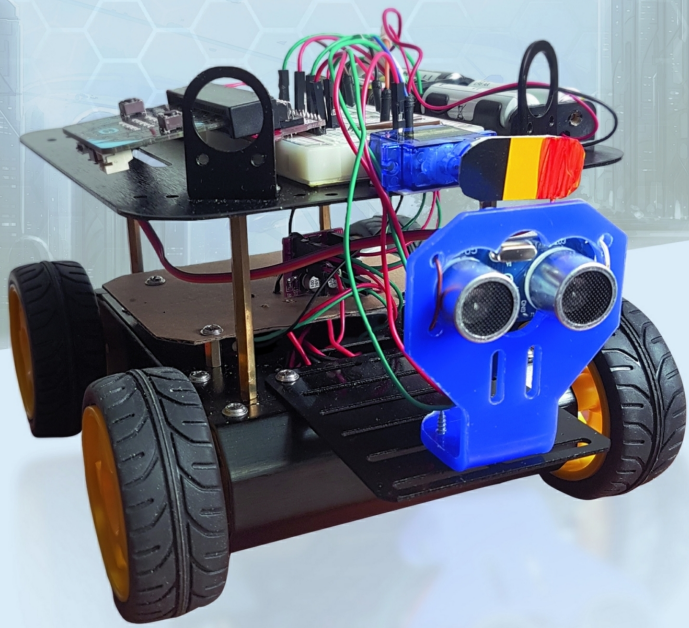
РОБОТОТЕХНИКА

ПРАКТИЧЕСКОЕ ВВЕДЕНИЕ ДЛЯ ДЕТЕЙ И ВЗРОСЛЫХ



Александр Фролов

Москва, 2020

6+

Александр Вячеславович Фролов

Робототехника: практическое введение для детей и взрослых

http://www.litres.ru/pages/biblio_book/?art=55547238
SelfPub; 2021
ISBN 978-5-532-95281-2

Аннотация

Эта книга поможет вам быстро освоить основы робототехники и приступить к конструированию собственных роботов, разных систем с микроконтроллерами и даже элементов умного дома. Вам не потребуются знания языков программирования и паяльник. Изучайте микроконтроллеры и робототехнику по нашей книге, и вы сможете быстро освоить современные профессии. Вы научитесь программировать micro:bit версий 1.5 и 2, работать с кнопками, светодиодами, светодиодным и OLED-экраном, измерять температуру, ускорение, напряженность магнитного поля. Используя интерфейс GPIO, вы подключите такие устройства, как моторы, сервоприводы, кнопки, датчики движения, расстояния и многие другие. Данные, полученные от контроллеров, вы сможете передавать на другие контроллеры с помощью радиоканала Bluetooth, а также контроллеров LoRa, способных обеспечить радиосвязь на расстояния, исчисляемые

километрами. Мы расскажем о том, как организовать электропитание робота или других конструкций.

Содержание

Александр Фролов
Робототехника: практическое введение для детей и взрослых

Содержание книги

Эта книга поможет вам быстро освоить основы робототехники и приступить к конструированию собственных роботов, систем автоматизации и даже элементов умного дома.

Если вы не программист, то наша книга позволит вам реализовать свои идеи в области робототехники с помощью визуальных средств разработки, не требующих программирования. Вам не потребуется паяльник – современные компоненты можно устанавливать на макетных платах, не вдыхая запах канифоли.

Изучайте микроконтроллеры и робототехнику по нашей книге, и вы сможете быстро освоить современные профессии.

Вы научитесь программировать micro:bit версий 1.5 и 2, научитесь работать с кнопками, светодиодами, светодиодным и OLED-экраном, измерять температуру, ускорение,

напряженность магнитного поля.

Используя интерфейс GPIO, предусмотренный на плате micro:bit, вы подключите такие устройства, как моторы, сервоприводы, обычные и сенсорные кнопки, датчики движения, расстояния и многие другие.

Данные, полученные от контроллеров, вы сможете передавать на другие контроллеры с помощью радиоканала Bluetooth, а также контроллеров LoRa, способных обеспечить радиосвязь на расстояния, исчисляемые километрами.

В отдельной главе мы расскажем о том, как организовать электропитание робота или других конструкций с микроконтроллерами.

В главе 1 «Запускаем микроконтроллер micro:bit» вы научитесь создавать простые программы для micro:bit версий 1.5 и 2 с помощью визуального редактора блоков Microsoft MakeCode for micro:bit.

Эти программы вы будете загружать в микроконтроллер через браузер, через сайт https://makecode.microbit.org/, через автономную программу Microsoft MakeCode, установленную на компьютере или ноутбуке, а также через приложение micro:bit, установленное в смартфоне или планшете.

Глава 2 «Управляем светодиодами» расскажем вам о том, как можно легко выводить на экран micro:bit, состоящий из 25 светодиодов, различную графическую, числовую и текстовую информацию.

Также с помощью генератора случайных чисел вы созда-

дите на этом экране простую анимацию – звездное небо и блуждающую точку.

В главе 3 «Работаем с кнопками» мы будем использовать кнопки A и B, расположенные на плате micro:bit, а также сенсорную кнопку в виде логотипа, доступную в micro:bit версии 2. Вы научитесь обрабатывать события, возникающие при нажатии этих кнопок, а также проверять их состояние во время работы программы. Кроме этого, вы сможете использовать контакты P0, P1 и P2 в качестве сенсорных кнопок.

Эти знания помогут вам управлять с помощью обычных и сенсорных кнопок работой программы звездного неба, созданной во второй главе, и другими различными программами.

В главе 4 «Измеряем температуру» вы научитесь пользоваться измерителем температуры, встроенным в процессор платы micro:bit. Это позволит вам создать программы, которые показывают текущую температуру процессора, а также контролировать превышение температуры сверх заданного значения.

Глава 5 «Измеряем ускорение» позволит вам контролировать движение платы micro:bit с ускорением или нахождение ее в состоянии невесомости. Вы научитесь обрабатывать жесты – поворот платы микроконтроллера логотипом вверх или вниз, наклон вправо или влево, поворот монитором вверх или вниз, встряхивание.

Все это будет возможно, когда вы научитесь работать с акселерометром, специально предназначенным для измерения ускорения во всех трех направлениях движения. Это устройство встроено в плату micro:bit.

Вы создадите программы, способные обнаружить невесомость и значительные перегрузки, реагировать на жесты, строить гистограмму значений ускорения.

Глава 6 называется «Встроенный магнитометр». В ней вы научитесь использовать магнитометр, расположенный на плате micro:bit, для исследования напряженности магнитного поля, а также сделаете простейший компас.

В главе 7 «Подключаем внешние устройства» мы научим вас работать с некоторыми внешними устройствами из набора DFRobot для micro:bit. Набор хорош тем, что не требует никаких навыков пайки и позволяет легко изучить способы подключения и программирования таких устройств, как светодиод, кнопки, моторы, потенциометры, датчики движения.

Вы сможете управлять яркостью небольшой светодиодной ленты и даже превратите свой micro:bit в простую музыкальную шкатулку.

Глава 8 «Осваиваем GPIO» посвящена подключению различных периферийных устройств к разъему GPIO микрокомпьютера micro:bit. Мы расскажем об этом разъеме и назначении его контактов, научим вас подключать к нему потенциометры, светодиоды, мощные лампочки. МЫ также

расскажем об использовании широтно-импульсной модуляции для управления яркостью светодиодов и лампочек, а также скоростью вращения моторов.

Вы подключите ультразвуковой и инфракрасный дальномеры, которые пригодятся при создании роботов, например, для исключения столкновений с препятствиями при движении.

Мы также расскажем об использовании макетных плат для сборки различных схем без помощи пайки.

Глава 9 «Запускаем радиоканал» будет полезна, если в вашем проекте используется несколько плат micro:bit. Для создания радиоканала между платами вы будете использовать контроллеры Bluetooth. Такое контроллер уже встроен в плату micro:bit.

Работая с этой главой, вы сделаете устройство с пультом управления, позволяющее включать и выключать дистанционно лампочку и вентилятор, а также научитесь передавать по радио данные телеметрии. Мы также расскажем, как управлять мощностью передаваемого сигнала, и как измерять мощность принятого сигнала.

Глава 10 «Подключаем micro:bit к компьютеру и второму micro:bit» научит вас передавать данные из micro:bit в терминальную программу, работающую на компьютере, подключенном к плате micro:bit через порт USB. Такая возможность будет очень удобна при отладке программ, работающих на micro:bit.

Также вы научитесь передавать данные между двумя платами micro:bit, соединенными двумя проводами с помощью интерфейса UART.

В главе 11 «Марсоход BoxRover заводит моторы» вы приблизитесь к созданию ровера – модели радиоуправляемого марсохода. Вы научитесь управлять моторами ровера с помощью транзистора и контроллера MX1508, реализующих функции H-моста и создадите программу, управляющую вращением двигателя.

Работая над этой главой, вы создадите первый прототип ровера BoxRover с микроконтроллером micro:bit, управляемого по радио со второго такого же контроллера.

В качестве домашнего задания вы сделаете простейшую систему передачи данных телеметрии, способную передавать в пульт управления температуру процессора micro:bit ровера. При этом на экране ровера будет отображаться температура процессора micro:bit пульта управления.

Глава 12 «Управляем сервоприводами» также посвящена управлению двигателями, только другого типа. В этой главе мы расскажем о сервоприводах, которые могут поворачивать свою ось на заданный угол, а также о сервоприводах непрерывного вращения, способных поддерживать скорость и направление вращения вала.

В этой главе вы узнаете, как устроены сервоприводы, а также научитесь создавать программы для micro:bit, способные ими управлять.

Вы создадите программу для ручного управление сервоприводом при помощи потенциометра, а в качестве домашнего задания – систему автоматического управления шлагбаумом. Она будет открывать шлагбаум, когда к нему приближается автомобиль, а после проезда автомобиля —автоматически закрывать шлагбаум.

В главе 13 «Осваиваем I2C» мы расскажем об интерфейсе I2C, специально предназначенном для подключения периферийных устройств к микроконтроллерам. На плате micro:bit уже есть все необходимое для работы с I2C.

Вы научитесь подключать к этому интерфейсу измеритель освещенности BH1750 FVI GY-30, погодную станцию Grove-BME280 или Grove-BMP280, а также OLED монитор, и, конечно, напишете программы для работы с ними.

На базе контроллера часов реального времени DS-3231 вы соберете часы с таймером и погодной станцией, позволяющие устанавливать текущую дату, текущее время, время срабатывания таймера. Когда таймер установлен, горит светодиод желтого цвета, а когда он сработал – белого. С помощью кнопок A и B, расположенных на плате micro:bit, вы сможете установить текущие дату и время, время срабатывания таймера, сбрасывать и устанавливать таймер.

Погодная станция будет показывать на экране монитора OLED температуру, давление, влажность и температуру точки росы.

Глава 14 «Обновление BoxRover» приблизит нас еще

на один шаг к созданию управляемого ровера.

Теперь наш прототип марсохода, управляемый по радио с помощью жестов, научится останавливаться перед препятствием при движении вперед и сигнализировать о такой остановке на экране монитора micro:bit, а также поднятием красного флажка с помощью сервопривода.

,Кроме всего этого, наш ровер будет отправлять в пульт управления телеметрические данные – температуру процессора и окружающей среды, давление, влажность, освещенность и температуру точки росы. Эти данные мы будем выводить на OLED-монитор, смонтированный в пульте управления.

В главе 15 «Электропитание робота» рассказано о том, какие существуют батарейки и аккумуляторы, как выбрать нужный тип питающих элементов для вашего робота. Будут рассмотрены особенности работы и зарядки аккумуляторов разных типов, применения контроллеров заряда и разряда Battery Management System (BMS).

Также вы узнаете, как использовать преобразователи и стабилизаторы для устройств, которым нужно разное напряжение питания, как подключать аккумуляторы и батарейки к макетной плате и устройствам робота с помощью отсеков и переходников, как питать робота от электрической сети 220 В на этапе отладки.

В главе 16 «Дальняя радиосвязь» рассмотрено использование технологии LoRa для создания каналов радио-

связи, способных передавать данные на большие расстояния, порядка километров и даже больше. При этом используются недорогие платы micro:bit и модули LoRa производства компании EBYTE.

Вы соберете погодную станцию, способную передавать информацию о температуре, давлении, влажности и температуре точки росы через канал дальней радиосвязи.

Вы также сделаете ретранслятор данных от сервера погодной станции, удваивающий максимальное расстояние передачи данных.

Глава 17 «Умный дом своими руками» содержит краткое введение в технологии умного дома. Мы расскажем о том, как использовать micro:bit и Raspberry Pi для сбора различных данных (о погоде, например) на обширной территории. Это может быть загородная резиденция или даже территория фабрики.

Почти в каждой главе предусмотрены домашние задания. Работая над ними самостоятельно, вы не только сможете проверить и закрепить свои знания. Решения домашних заданий вы сможете найти на сайте автора этой книги.

Исходные коды программ

Исходные коды всех программ, опубликованных в книге, вы можете скачать на сайте автора http://frolov-lib.ru/books/boxrover/, а также на GitHub https://github.com/AlexandreFrolov/BoxRover.

Свои пожелания, замечания и предложения вы можете оставить в группе по адресу https://www.facebook.com/groups/539933346894981/, а также отправить по электронной почте на адрес microbit@frolov.pp.ru.

Введение

Каждый день вы используете различную технику, даже не задумываясь о том, что в ней есть встроенные микрокомпьютеры. Вещи, которыми мы пользуемся в обиходе, становятся все более интеллектуальными.

Робот-пылесос в процессе уборки сам обследует квартиру, составляя ее карту, сам возвращается на пункт подзарядки, и сообщает о своих действиях голосом. Роботы моют окна, работают на выставках и барах, управляют автомобилями, используются в боевых действиях и в борьбе с терроризмом.

Новостные сайты в интернете забиты статьями о том, что скоро роботы начнут отнимать работу у людей, вытесняя их из ряда профессий, в том числе не требующих высокой квалификации. С этим можно спорить, однако несомненно, что робототехника уже прочно вошла в нашу жизнь, и специалисты в этой области будут всегда востребованы.

Но с чего лучше всего начинать обучение программированию и робототехнике?

Можно условно выделить два подхода к обучению.

Первый подход предполагает изучение основ электроники, компьютерной техники, программирования на ассемблере и Си, с последующим переходом к более высокоуровневым средствам проектирования программ и схемотехниче-

ским решениям на базе современных процессоров и микроконтроллеров, обучения основам численных методов и дискретной математики.

Когда-то давным-давно был доступен только первый способ, и я сам обучался именно так. Я собирал приемники-передатчики на транзисторах и лампах, различные радио устройства, триггеры из транзисторов, регистры из элементов 2И-НЕ на базе К155ЛА3, электронные часы, частотомер и самодельные компьютеры на микросхеме КР580ВМ80А, добавляя россыпь логических микросхем, микросхем памяти и периферийных устройств (рис. В.1). Первые программы были написаны, конечно же, в машинных кодах!

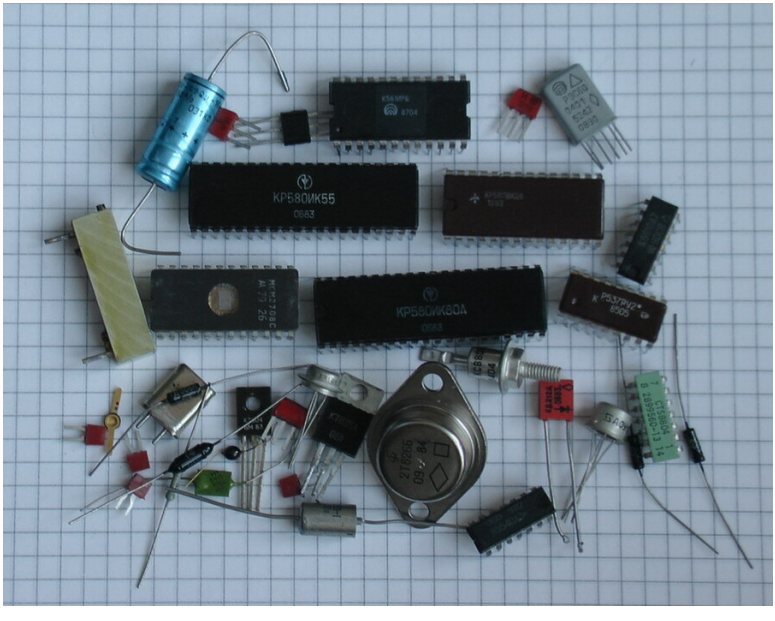


Рис. В.1. Элементная база прошлого века

У этого подхода есть очевидные преимущества и не менее очевидные недостатки.

Из важных преимуществ – на любом этапе будет понятна суть происходящего на самом низком уровне, на уровне сигналов и команд процессора, на уровне элементов и микросхем. Понимание сложных вещей будет достигаться постепенно, по мере продвижения от простого к сложному.

В то же время огромный недостаток метода обучения с самых основ заключается в том, что весь процесс отнимет

очень много времени и сил. Двигаясь с самого начала, вы не скоро сможете создать что-нибудь достаточно сложное и интересное. В то же время хочется как можно быстрее насладиться результатами своего труда.

Второй подход предполагает, что обучение начинается с использования готовых наборов, содержащих микроконтроллеры и периферийные блоки, с применением графических инструментов программирования и языков высокого уровня. Далее после достижения определенных результатов, можно перейти к изучению низкоуровневых средств и инструментов. Именно этот подход мы и будем использовать в данной книге.

Проект модели марсохода BoxRover

На мой взгляд, в процессе обучения нужно двигаться к какой-нибудь интересной, но сложной цели, при этом шаги должны быть достаточно крупными, чтобы продвижение было заметно. Сложность цели необходима для получения самых разносторонних знаний, которые пригодятся в дальнейшем на работе в области ИТ и робототехники.

Здесь интересно было бы создать так называемый STEM-проект, реализация которого позволить получить знания, необходимые в реальной жизни. Аббревиатура STEM – это сокращение от Science, Technology, Engineering и Math, т.е. наука, технология, инженерное дело и математика.

Я предлагаю в качестве такого учебного проекта создать несложную модель марсохода (или движущегося робота для изучения каких-либо других планет) с названием BoxRover. Пусть ваше устройство никогда не полетит в космос, но оно сможет управляться по радио или через интернет, измерять различные параметры окружающей среды, получать фотографии и видео, и передавать все это «на землю», например, в ваш компьютер или планшет.

По мере создания модели марсохода BoxRover вы научитесь программировать встроенные микроконтроллеры, управлять движением, получать данные бортовых измерительных устройств и получите другие необходимые знания.

На рис. В.2 мы показали пример современных компонентов, из которых вы будете делать своего первого робота.

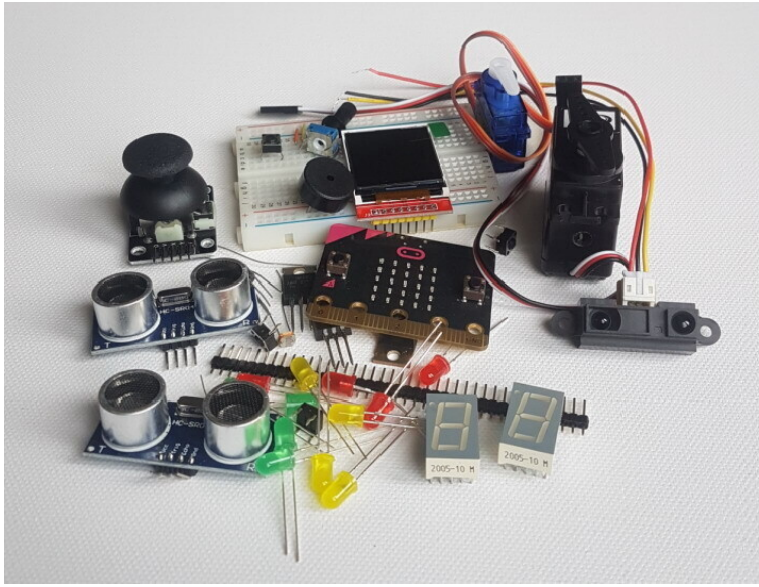


Рис. В.2. Некоторые современные компоненты для изготовления робота

На следующем этапе вы сделаете и другие проекты, например, элементы автоматизации умного дома.

Выбор платформы для обучения

На различных курсах робототехники для детей используются наборы Lego. С их помощью можно быстро собрать робота из готовых деталей и так же быстро запрограммировать на выполнение различных несложных действий. На мой взгляд, такие наборы, хотя и дают представление о некоторых функциях роботов, все же недостаточно хорошо демонстрируют взаимодействие микрокомпьютеров и периферийных устройств.

Чтобы лучше понимать, что происходит, научиться не только программировать готового робота, но и создавать собственные проекты из электронных блоков и компонентов, на разных стадиях обучения мы будем работать с микроконтроллерами BBC micro:bit, платформой Arduino, STM32 Nucleo, а также Raspberry Pi.

Микроконтроллер BBC micro:bit

Микроконтроллер BBC micro:bit был создан корпорацией BBC как открытый проект, нацеленный на повышение компьютерной грамотности, и в первой версии стал доступен в феврале 2016 года. Сейчас в продаже появилась значительно улучшенная версия 2 этого микроконтроллера.

В России micro:bit версии 1.5 можно купить в нескольких интернет-магазинах по цене ниже 1900 руб., что делает его весьма доступным решением для начала обучения. На момент написания книги версию 2 можно было приобрести в магазине https://mrobot.by (примерно по такой же цене), а также в зарубежных магазинах. На странице https://microbit.org/buy/ представлен список компаний из разных стран, где можно купить micro:bit версии 2.

Даже первая версия этого недорогого микрокомпьютера размером с половину кредитной карты оснащена неплохим набором периферийных устройств. В micro:bit версии 2 был добавлен микрофон, динамик, еще одна сенсорная кнопка, увеличен объем памяти и мощность процессора. Кроме того, появился режим сохранения энергии, что важно при питании от батарей и аккумуляторов.

В табл. 1 вы найдете сравнение характеристик micro:bit версии 1.5 и 2 с другими широко распространенными микроконтроллерами – Arduino UNO и STM32 Nucleo F401RE.

Табл. B1. Сравнение характеристик микроконтроллеров

Параметр	Micro:bit v1.5	Micro:bit v2	Arduino UNO	STM32 Nucleo F401RE
Рабочая частота, МГц	16	64	16	84
Разрядность, бит	32	32	8	32
Объем ОЗУ, Кбайт	16	128	2	96
Процессор	nRF51822	nRF52833 с модулем FPU	ATmega 328	STM32 F401E
Объем памяти Flash, Кбайт	256	512	2	512
Кол-во цифровых контактов	19	19	20	81
Микрофон	-	1	-	-
Динамик	-	1	-	-
Кнопки	2	3	-	-
Кол-во аналоговых входов	3	3	6	10
Разрядность АЦП	10	10	10	12
Интерфейсы SPI	1	1	1	4
Интерфейсы I2C	1	1	1	3
Интерфейсы UART	1	1	1	3
Таймеры	-	-	3	10
Напряжение питания, В	3,3	3,3	7–12	5, 7-12

Если вы создаете проект робота или какой-либо другой проект с микроконтроллером, то при использовании micro:bit v2 можете воспользоваться многими устройствами, установленными на плате этого микроконтроллера:

- 32-разрядный процессор Nordic nRF52833 Arm Cortex-M4 с тактовой частотой 64 МГц и встроенным математический сопроцессором FPU (Floating Point Unit), ускоряющим выполнение операций над вещественными числами;
- Flash-память объемом 512 Кбайт, содержимое которой сохраняется при выключении питания;
- оперативная память объемом 128 Кбайт;
- три программируемые кнопки, одна из которых сенсорная;
- монитор из 25 светодиодов;
- микрофон;
- динамик;
- акселерометр (датчик ускорения);
- магнитометр (встроенный компас);
- измеритель температуры;
- встроенный модуль Bluetooth диапазона 2,4 ГГц для беспроводных коммуникаций;
- интерфейс USB;
- встроенный аналого-цифровой преобразователь;
- интерфейс GPIO (General Purpose Input Output)

Заметим, что в micro:bit v2, в отличие от v1.5, шина I2C полностью выделена для внешних устройств. К ней не подключены устройства, расположенные на плате micro:bit.

Также добавился четвертый свободный для использования контакт интерфейса GPIO. Была увеличена допустимая токовая нагрузка на подключаемую периферию. Если раньше в micro:bit v1.5 суммарный ток на контактах GPIO не должен был превышать 90 мА, то в версии 2 этот предел расширен до 190 мА.

Как видите, в микроконтроллере micro:bit уже имеется встроенное оборудование, которое потребуется нам для модели марсохода (рис. В.3, В.4). Устройства, которые появились на плате micro:bit v2 показаны на рис. В.5 и В.6.

Даже с базовым оборудованием micro:bit версии 2 наш марсоход сможет измерять температуру, напряженность магнитного поля, реагировать на ускорения, подавать звуковые сигналы, обнаруживать источники звука, и даже показывать инопланетным зрителям картинки на мониторе!

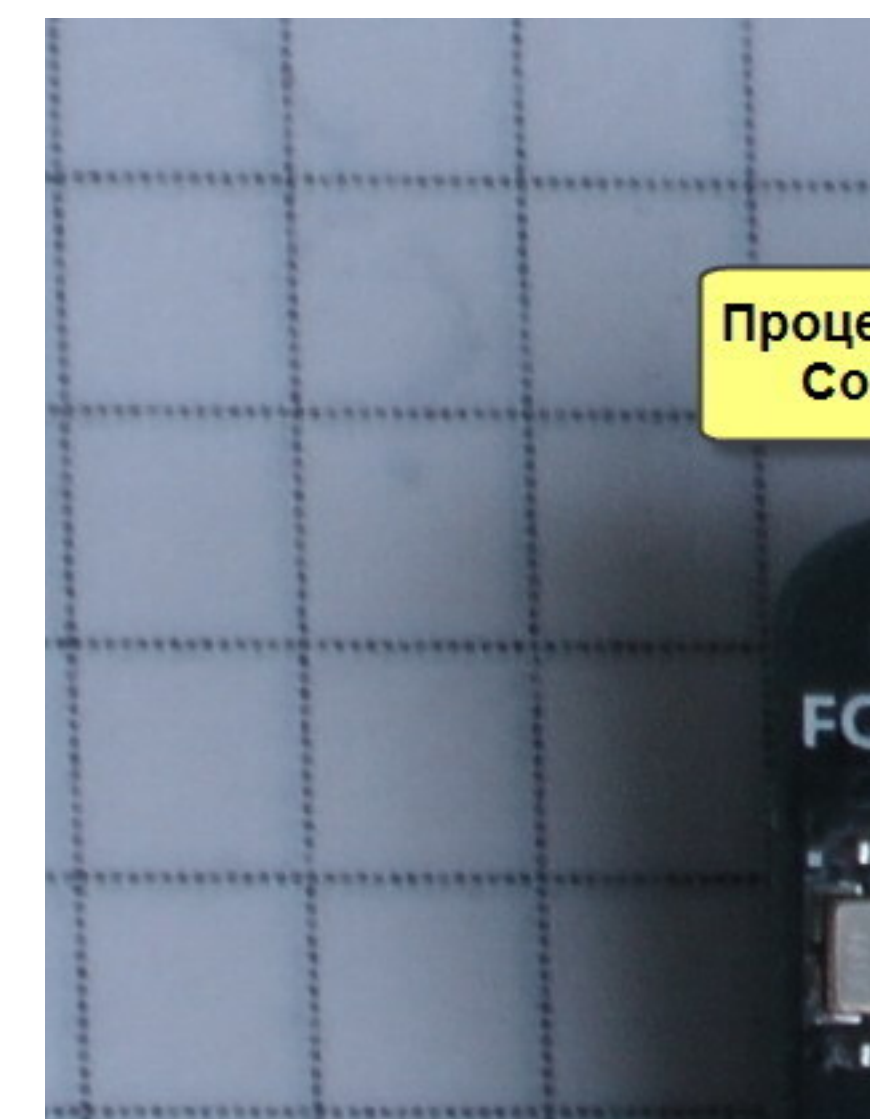

Проце
Со
FC

Рис. В.3. Оборудование на плате micro:bit v1.5

По мере реализации проекта BoxRover мы подключим к micro:bit и другие устройства, например, контроллеры, предназначенные для управления двигателями платформы, погодную станцию, OLED-монитор.

Особенно следует отметить наличие у micro:bit интерфейса GPIO. Используя порты GPIO, вы сможете подключать к микроконтроллеру различные цифровые и аналоговые устройства.

Два контакта порта GPIO используются для подключения внешних устройств, работающих с протоколом I^2C (Inter-Integrated Circuit). Также предусмотрено три контакта для обмена данными с устройствами по протоколу SPI (Serial Parallel Interface).

Наличие портов I^2C (встречается обозначение I2C) и SPI дает возможность подключить к микроконтроллеру такие устройства, как моторы и шаговые двигатели, датчики движения, датчики газа и наличия воды, радио модули и т.д. Эти устройства продаются в интернет-магазинах и, как вы увидите, легко подключаются к micro:bit и программируются.

Микрокомпьютер micro:bit при питании от батарей потребляет всего несколько десятков мА при напряжении питания 3В. И это если включены все светодиоды, а процессор загружен на полную мощность. Заметим, что макет нашего марсохода будет питаться от батарей или аккумуляторов, поэтому важно, чтобы все бортовые устройства потребляли как

можно меньше энергии.

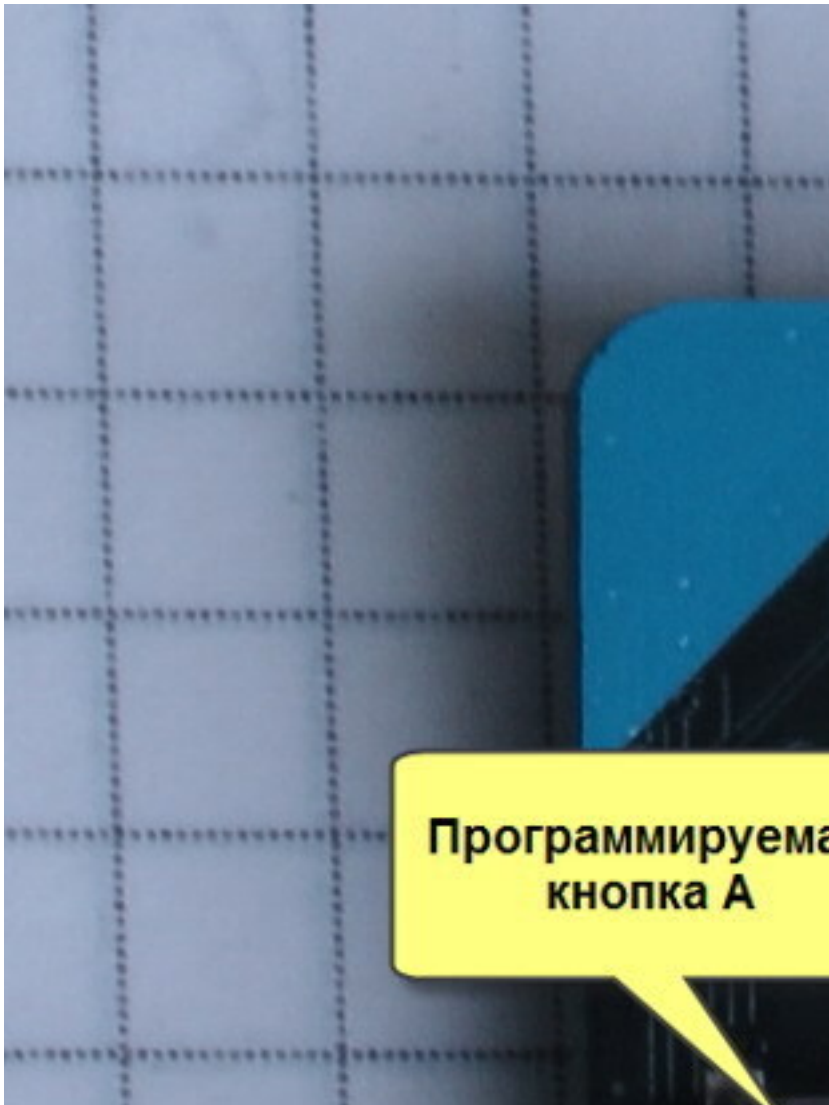

Программируема
кнопка A

Рис. В.4. Кнопки и светодиодный экран на плате micro:bit v1.5

Начинающие программисты смогут воспользоваться визуальным редактором программ MakeCode. Этот редактор доступен через браузер, так что его даже не надо устанавливать на компьютер (хотя есть и версия для установки на Windows и MacOS). Также можно работать с MakeCode при помощи мобильного приложения, доступного для смартфонов и планшетов на базе Android и iOS.

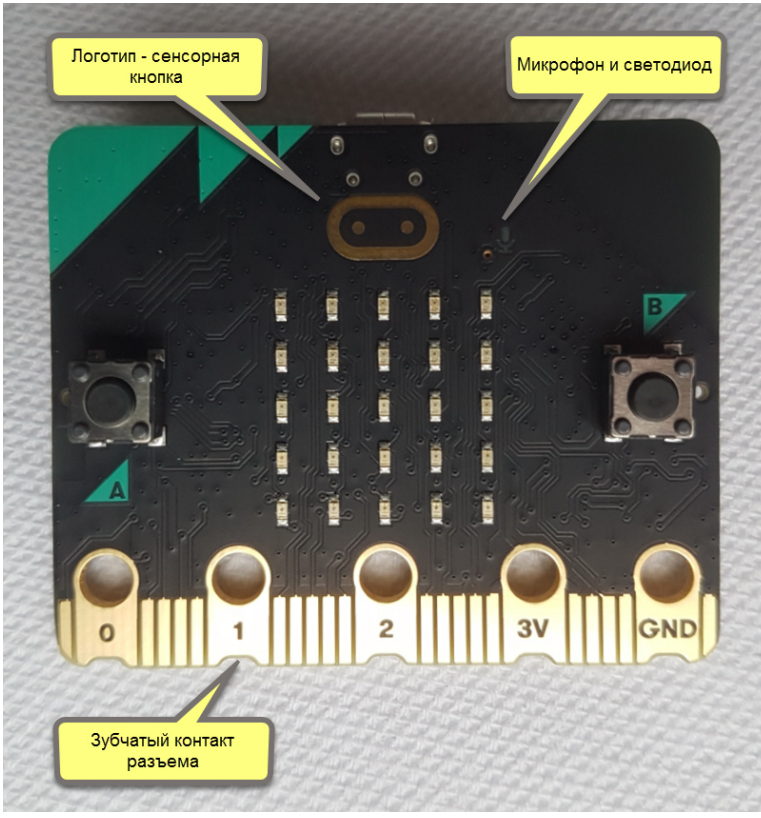


Рис. В.5. Сенсорная кнопка в виде логотипа, микрофон и светодиод на плате micro:bit v2

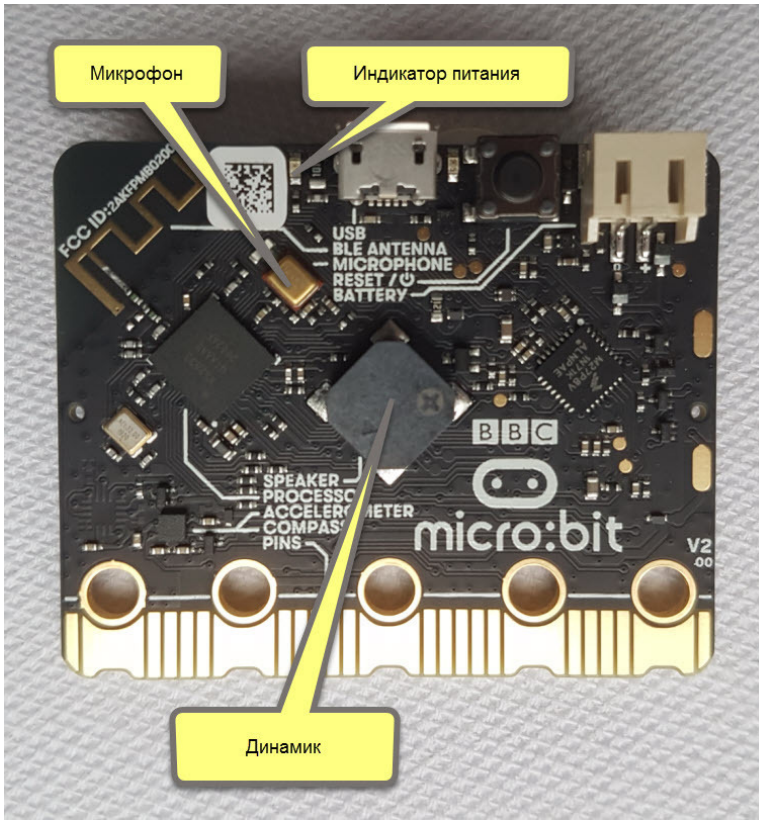


Рис. В.6. Микрофон, индикатор питания и динамик на плате micro:bit v2

Для micro:bit можно создавать программы на языке Python, используя несложный в использовании редактор

Python Editor. Есть также инструменты, позволяющие программировать micro:bit на языках JavaScript, C и даже Ada.

Микроконтроллер Arduino

Микроконтроллер Arduino был создан в 2005 году для быстрого обучения студентов работе с электронными проектами. Сейчас он стал очень популярен, однако, на мой взгляд, Arduino имеет заметно более высокий порог вхождения по сравнению с BBC micro:bit.

Прежде всего, для программирования Arduino используется не самый простой в изучении язык C++. Вам также придется установить на свой компьютер Arduino IDE, либо использовать онлайн редактор Arduino Web Editor.

Изучение языков C и C++ очень полезно в плане профессиональной работы с микроконтроллерами, однако на первом этапе при использовании micro:bit можно получить представление о робототехнике и без знаний этих языков программирования.

Далее, на плате Arduino нет таких устройств, как на micro:bit. Предполагается, что для подключения различной периферии (датчиков, например) вы будете приобретать платы расширения (Arduino Shield или шилды). Эти платы и устройства можно подключить к плате Arduino через разъем GPIO.

В продаже имеется очень много плат расширения Arduino Shield самого разного назначения, однако все их нужно покупать дополнительно.

Что касается энергопотребления, то сам по себе микрокомпьютер Arduino довольно экономичен. Плата Arduino Uno требует питание 9 В, потребляя при этом порядка 50 мА. Есть и более экономный вариант – Arduino Pro Mini. Он может использовать для питания напряжение 3,3 В (как micro:bit), и при этом потребляет всего несколько десятков мкА. Тем не менее, следует учитывать энергопотребление дополнительных модулей (как и в случае micro:bit), а оно может быть довольно значительным.

Конечно, платформу Arduino можно использовать для создания своей модели марсохода и других проектов, однако мы начнем с платформы BBC micro:bit, как более легкой в освоении.

Платформа STM32 Nucleo F401RE

В то время как Arduino представляет собой платформу для обучения и любительских разработок, в профессиональной области большое распространение получили микроконтроллеры STM32 производства STMicroelectronics https://www.st.com/.

Платформа STM32 Nucleo F401RE, совместимая с модулями расширения Arduino, представляет намного больше возможностей по сравнению с платформой Arduino и micro:bit.

С точки зрения применения в робототехнике у микроконтроллеров на базе STM32 намного больше возможностей, чем у Arduino или micro:bit.

Однако все имеет свою цену. И если роботы на базе micro:bit можно разрабатывать с применением очень простых средств блочного визуального программирования, то при использовании Arduino и STM32 не обойтись без знаний языков программирования C или C++. Кроме того, архитектура процессоров STM32 достаточно сложна, а в полном руководстве по моделям STM32 насчитывается более 1700 страниц!

Тем не менее, вам не придется читать это руководство на первом этапе знакомства. Имеются средства программирования для STM32, скрывающие сложность внутри готовых

библиотек и программных модулей.

Микрокомпьютер Raspberry Pi

Для решения серьезных задач, таких как обработка данных, полученных от измерительной аппаратуры нашей модели марсохода, передачи видео «на землю», распознавание изображений, передачи данных, полученных от контроллеров умного дома через интернет и т.п. микрокомпьютеры BBC micro:bit и Arduino, к сожалению, не подойдут. В то же время относительно недорого можно приобрести миниатюрный одноплатный компьютер Raspberry Pi, способный составить конкуренцию в некоторых случаях даже настольным компьютерам.

Например, модель Raspberry Pi 4 содержит 64-разрядный 4-ядерный процессор ARMv8-A с тактовой частотой 1,5 Ггц. У него есть встроенный графический процессор GPU Broadcom VideoCore VI, беспроводные интерфейсы Bluetooth и WiFi, разъем RJ-45 Ethernet с пропускной способностью 1 Гбит, интерфейс видеокамеры, два интерфейса Micro HDMI, интерфейсы USB и другое оборудование.

К микрокомпьютеру Raspberry Pi через шину GPIO можно подключить самые разнообразные устройства, такие как измерители параметров, двигатели и пр.

Надо понимать, что для Raspberry Pi 4 нужен довольно мощный источник электропитания. Штатный блок питания, например, обеспечивает 3 А при напряжении 5 В, а это уже

15 Вт. Если устанавливать этот микрокомпьютер на макет марсохода, то для его питания (а также для питания дополнительных устройств) в течение длительного времени потребуются довольно тяжелые и емкие аккумуляторы. Будет нужно прочное шасси и мощные электродвигатели, которые сами по себе будут потреблять много электроэнергии.

Вы можете сделать на базе Raspberry Pi 4, например, сервер обработки данных, полученных от макета марсохода или контроллеров умного дома, Web-сервер для трансляции этих данных в интернете и для решения других подобных задачи. Вы можете создавать программы для Raspberry Pi с использованием практически любых языков программирования, доступных для обычных серверов, настольных компьютеров и ноутбуков.

Другие микрокомпьютеры

На момент создания книги на рынке появилось очень много одноплатных микрокомпьютеров различного типа и назначения.

Это многочисленные «клоны» Arduino и Raspberry Pi, которые отличаются ценой, габаритами и потребляемой мощностью. Есть даже готовая мощная система NVIDIA Jetson Nano в миниатюрном исполнении, способная решать серьезные задачи искусственного интеллекта, компьютерного зрения и робототехники.

В продаже можно найти микрокомпьютеры с очень небольшим энергопотреблением, что будет полезно для создания нашего макета или для устройств интернета вещей IoT (Internet of Things).

1. Запускаем микроконтроллер micro:bit

Микроконтроллер micro:bit можно купить либо просто в виде платы, либо в составе набора. В минимальном варианте набор состоит из платы контроллера, корпуса для двух батареек AAA с соединительным проводом и USB-кабеля, с помощью которого можно подключить micro:bit к ноутбуку или настольному компьютеру.

В продаже также есть различные наборы модулей и аксессуаров, расширяющих возможности микроконтроллера micro:bit, и даже готовые наборы для сборки различных роботов. На первом этапе вам достаточно будет приобрести только плату micro:bit или самый простой набор.

Первое включение

При первом включении на micro:bit запускается демонстрационная программа. Мы заменим ее нашей программой. Чтобы наша программа заработала на micro:bit, ее нужно создать с помощью того или иного инструмента, а затем загрузить в память микроконтроллера.

Программы можно загружать в micro:bit с помощью ноутбука или стационарного компьютера, а также с помощью смартфона или планшета на базе Android или iOS. Удобнее всего работать с ноутбуком или компьютером, поэтому мы сначала рассмотрим именно такой вариант.

Итак, подключите микроконтроллер micro:bit к USB-порту ноутбука или компьютера с помощью переходника USB – микро USB (рис. 1.1). Если такой кабель не водит в набор, то вы можете приобрести его отдельно.

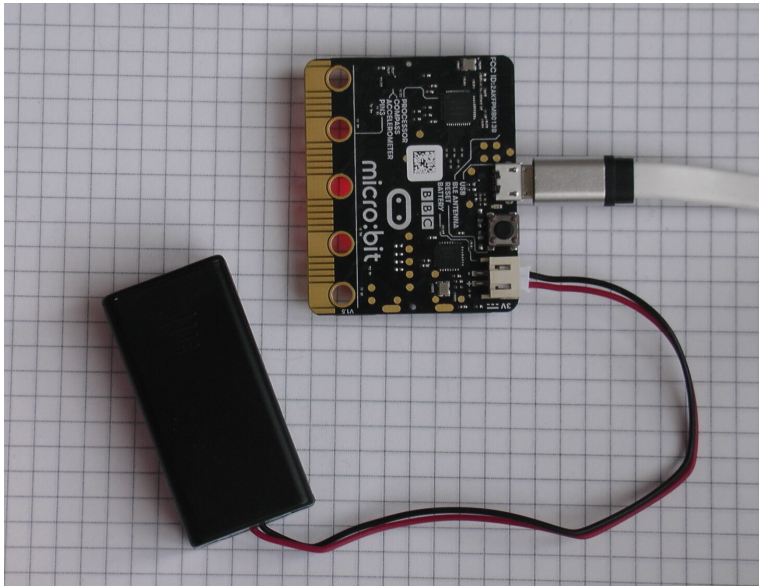


Рис. 1.1. Первое включение micro:bit

После подключения на экране micro:bit появится бегущая строка HELLO. Это означает, что micro:bit работает нормально, и в нем запустилась программа, установленная по умолчанию. Нашей задачей на данном этапе будет замена этой программы собственной, созданной с помощью программы Microsoft MakeCode for micro:bit.

Если micro:bit подключен к компьютеру через порт USB, то ему не требуется никакого дополнительного питания. Для автономной работы после загрузки в память контроллера

нужной программы отключите кабель USB и подключите блок батарейного питания в специально предназначенный для этого разъем (рис. 1.1).

Если вам проще работать с micro:bit, не загружая на свой компьютер никакие программы, читайте ниже в этой главе раздел «Программирование в браузере».

Чтобы загрузить автономную версию программы Microsoft MakeCode for micro:bit, откройте сайт https://makecode.microbit.org/offline-app. В нижней части страницы отметьте флажок **I agree to these Microsoft Software License Terms and to the Microsoft Privacy Statement.** После этого вы увидите ссылки на скачивание (рис. 1.2).

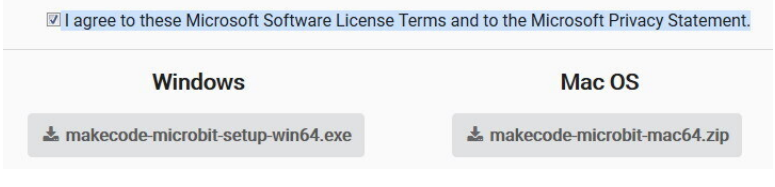


Рис. 1.2. Загрузка автономной версии программы Microsoft MakeCode for micro:bit

Если вы работаете с компьютером на базе Windows, скачайте программу makecode-microbit-setup-win64.exe. Если же на ваш компьютер установлена Mac OS, вам потребуется программа makecode-microbit-mac64.zip.

Программа makecode-microbit-setup-win64.exe не требует

никакой установки и начинает работать сразу после запуска (рис. 1.3).

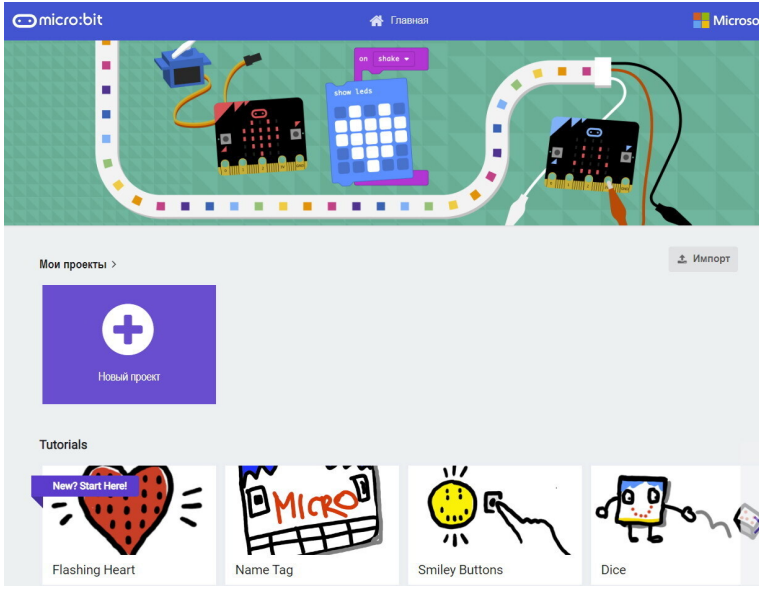


Рис. 1.3. Главное окно программы Microsoft MakeCode for micro:bit

Щелкните в главном окне кнопку **Новый проект** или **New Project** (если все надписи показаны на английском языке). Вы увидите окно нового проекта, где мы и будем программировать наш микроконтроллер (рис. 1.4).

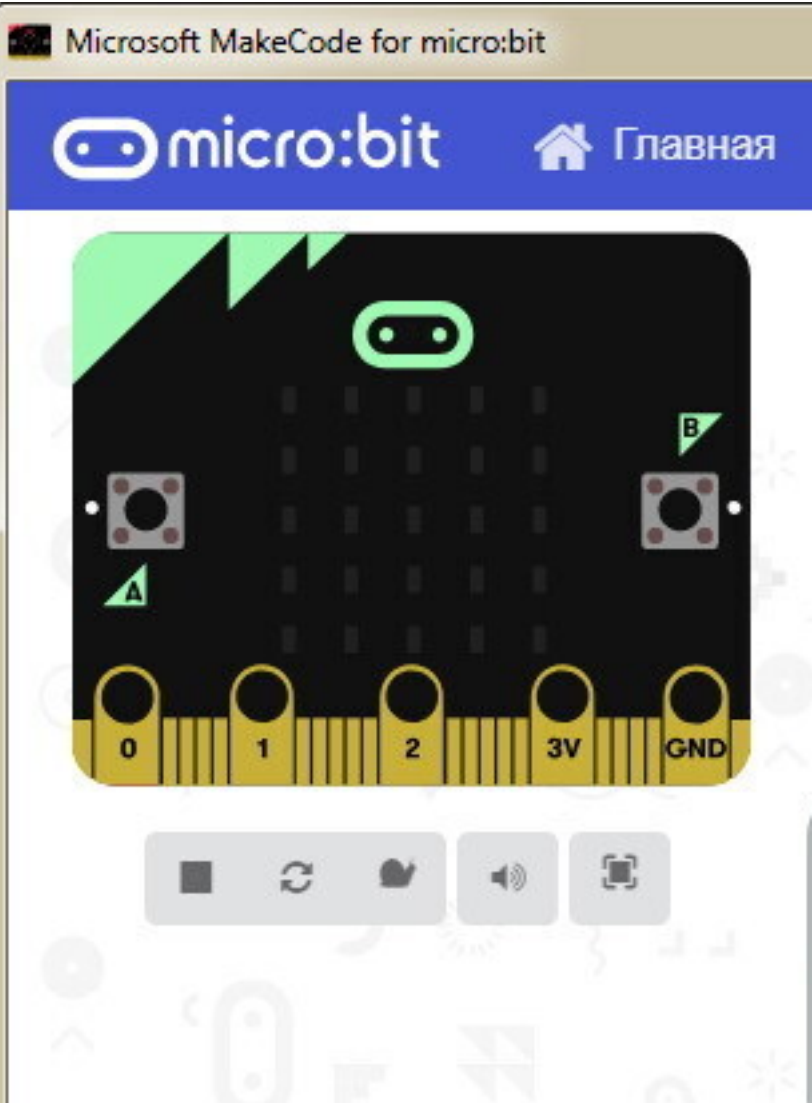

Microsoft MakeCode for micro:bit
micro:bit
Главная
B
A
0
1
2
3V
GND

Рис. 1.4. Создание нового проекта

При необходимости укажите, на каком языке будут показываться все надписи в программе. Для этого щелкните изображение шестеренки в правом верхнем углу окна программы, и выберите строку **Язык** или **Language**. Затем щелкните название нужного вам языка.

Давайте посмотрим внимательно на главное окно только что созданного нами проекта.

Слева вверху находится изображение платы контроллера micro:bit, под которым расположены кнопки отладки нашей программы. С помощью этих кнопок вы можете создавать и отлаживать программы, даже не подключая контроллер micro:bit к компьютеру.

В средней части окна находятся палитры блоков, из которых можно составить программу. На рис. 1.5 мы раскрыли палитру **Основное**.

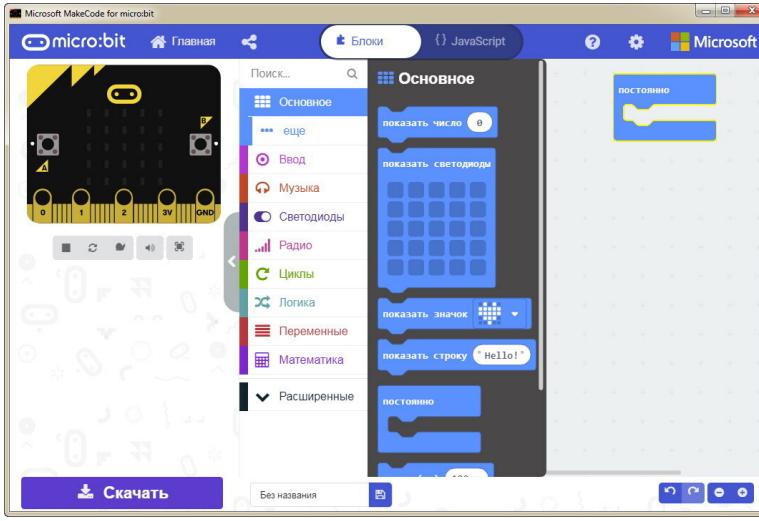


Рис. 1.5. Палитра компонентов Основное

Перетащите мышью блок **показать светодиоды** вправо, в блок **постоянно**. Далее проделайте эту операцию еще раз, расположив второй такой же блок под первым (рис. 1.6).

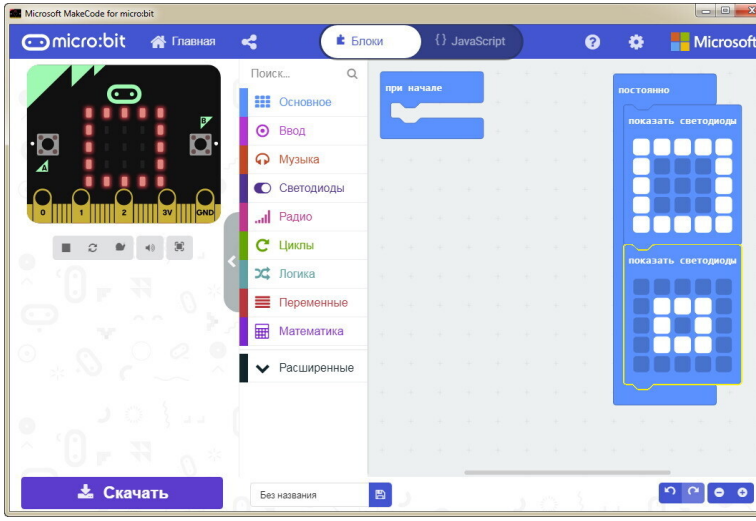


Рис. 1.6. Добавлены два блока показа светодиодов

Щелкая мышью изображения светодиодов, создайте в верхнем блоке квадрат большого размера, а в нижем – маленького, как это показано на рис. 1.6.

Теперь щелкните кнопку **Скачать**. После этого подготовленная нами программа загрузится в память микроконтроллера и сразу запустится на выполнение. Вы увидите, что на экране micro:bit будет попеременно отображаться то большой, то маленький квадрат (рис. 1.7).

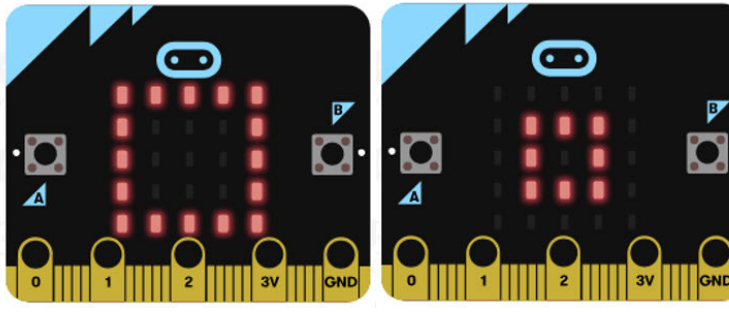


Рис. 1.7. Наша программа работает на микроконтроллере

Справа от кнопки **Скачать** есть поле, в котором вы можете ввести название программы. А еще правее находится кнопка с изображением дискеты, предназначенная для сохранения программы в виде файла с расширением имени hex.

В микроконтроллере micro:bit версии 2 используется новый формат файла hex с названием Universal Hex. Файлы с этим форматом могут работать и на micro:bit версии 1.5. Обратное неверно – файлы hex старого формата можно использовать на micro:bit версии 2 только если загрузить их в редактор, а потом сохранить. При такой операции выполняется изменение формата.

Всегда сохраняйте программы, над которыми работаете, иначе есть риск потерять время, например, при случайном отключении питания компьютера или при каких-либо оши-

бочных действиях.

При подключении micro:bit через USB-порт в компьютере появляется дисковое устройство. Но это не настоящий диск. Если скопировать hex-файл на такой «диск», то программа загрузится в micro:bit и автоматически запустится на выполнение. После запуска она будет удалена с диска, созданного при подключении контроллера.

Итак, мы создали и запустили на выполнение нашу первую программу, которая в цикле показывает два изображения на экране микроконтроллера. Правда, это было несложно?

Теперь вы можете отсоединить micro:bit от порта USB и подключить батарейное питание. Загруженная нами программа продолжит свою работу – она остается в памяти микроконтроллера даже при отключении электропитания.

Напомним, что все программы из нашей книги можно загрузить на сайте автора по адресу http://frolov-lib.ru/books/boxrover/ в виде zip-архива. Программа, над которой вы только что работали, называется **Квадрат**, и ее можно найти в каталоге BoxRover/ch01/microbit-Квадрат.hex архива.

Программирование в браузере

Если нет возможности или желания скачивать и устанавливать на компьютер Microsoft MakeCode for micro:bit, то вы можете программировать micro:bit с помощью браузера, открыв MakeCode Editor на сайте https://makecode.microbit.org/, а также с помощью смартфона или планшета. В этом разделе мы расскажем об использовании браузера.

Откройте сайт, и вы увидите страницу, очень похожую на главное окно программы Microsoft MakeCode for micro:bit (рис. 1.8).

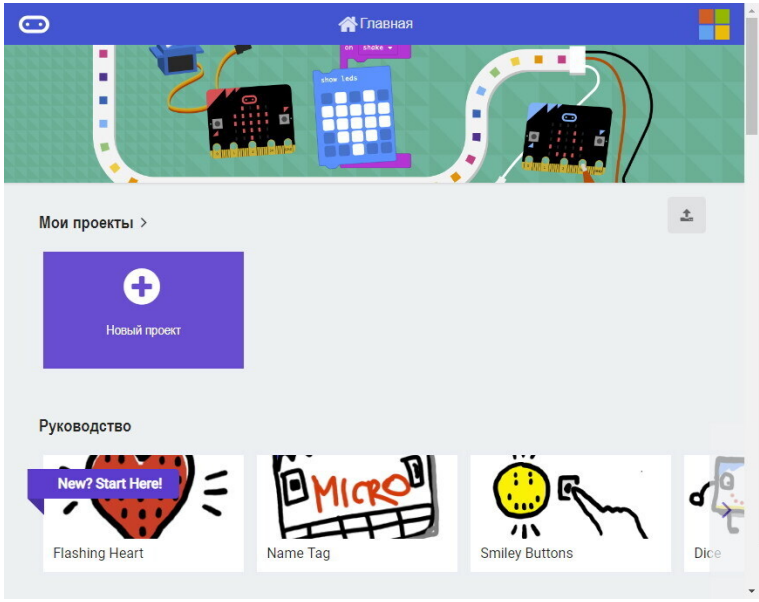


Рис. 1.8. Сайт makecode.microbit.org

Повторите действия, которые мы проделали с этой программой, чтобы создать такую же программу.

Затем щелкните кнопку **Скачать**, и после завершения загрузки скопируйте полученный hex-файл на устройство, переместив его значок из папки загрузки на значок устройства micro:bit (рис. 1.9).

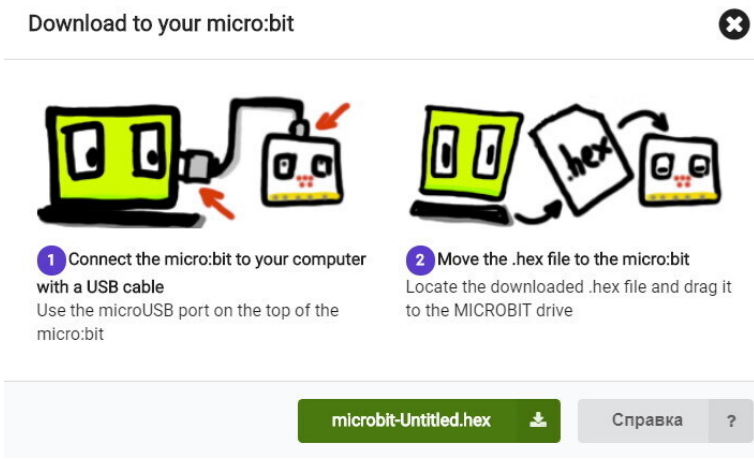


Рис. 1.9. Скачивание двоичного кода программы с сайта makecode.microbit.org

Программа Microsoft MakeCode for micro:bit удобнее тем, что не требует скачивания и копирования hex-файла программы. Она сразу загружает файл в устройство, после чего программа начинает работать. В результате вы будете быстрее создавать и отлаживать программы.

Создаем программы для micro:bit в смартфоне

Если у вас нет ноутбука или настольного компьютера, то для программирования micro:bit можно использовать смартфон или планшет. При этом программы вы будете загружать в micro:bit при помощи беспроводного интерфейса Bluetooth.

Заметим, что с планшетом работать удобнее, т.к. у него больше размер экрана.

Прежде всего, вам нужно установить на свой планшет или смартфон приложение micro:bit. Если у вас устройство на базе Android, ищите это приложение в Google Play, а если вы пользуетесь iPhone или iPad, то вам нужен App Store.

Прежде чем вы сможете загружать программы со смартфона или планшета на micro:bit, необходимо включить на вашем мобильном устройстве Bluetooth.После включения аккумулятор будет расходоваться заметно быстрее, так что не забудьте выключить Bluetooth после завершения работы с micro:bit.

Итак, вы установили на смартфон приложение и разрешили работу Bluetooth. Теперь нужно установить соединение между смартфоном и micro:bit. К сожалению, эта процедура не такая простая, как хотелось бы. Она состоит из нескольких шагов.

Прежде всего, отключите micro:bit от USB-интерфейса компьютера, если он был подключен, и переведите micro:bit на питание от батареи. Иначе соединение установить не получится.

Далее запустите приложение micro:bit и дотроньтесь до кнопки **Connect** (рис. 1.10).

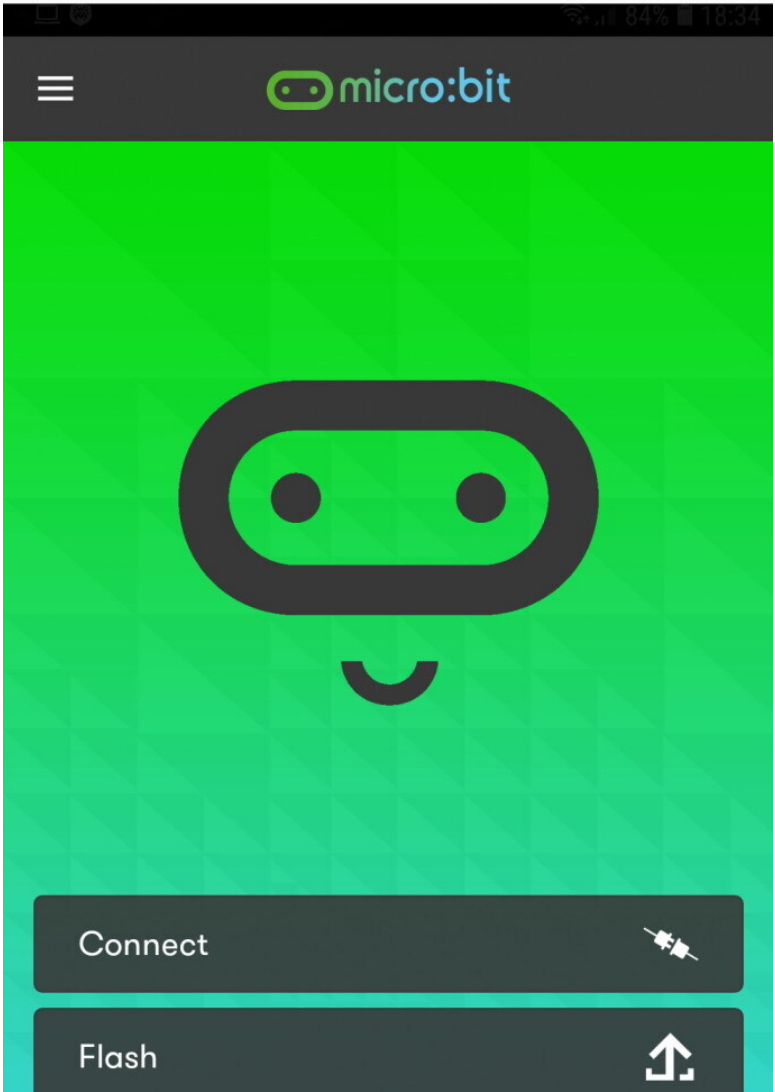

micro:bit
Connect
Flash

Рис. 1.10. Главное окно приложения micro:bit

В следующем окне приложения используйте кнопку **Pair a new micro:bit**. На рис. 1.11 показан случай, когда вы ранее не подключали к смартфону микрокомпьютер micro:bit.

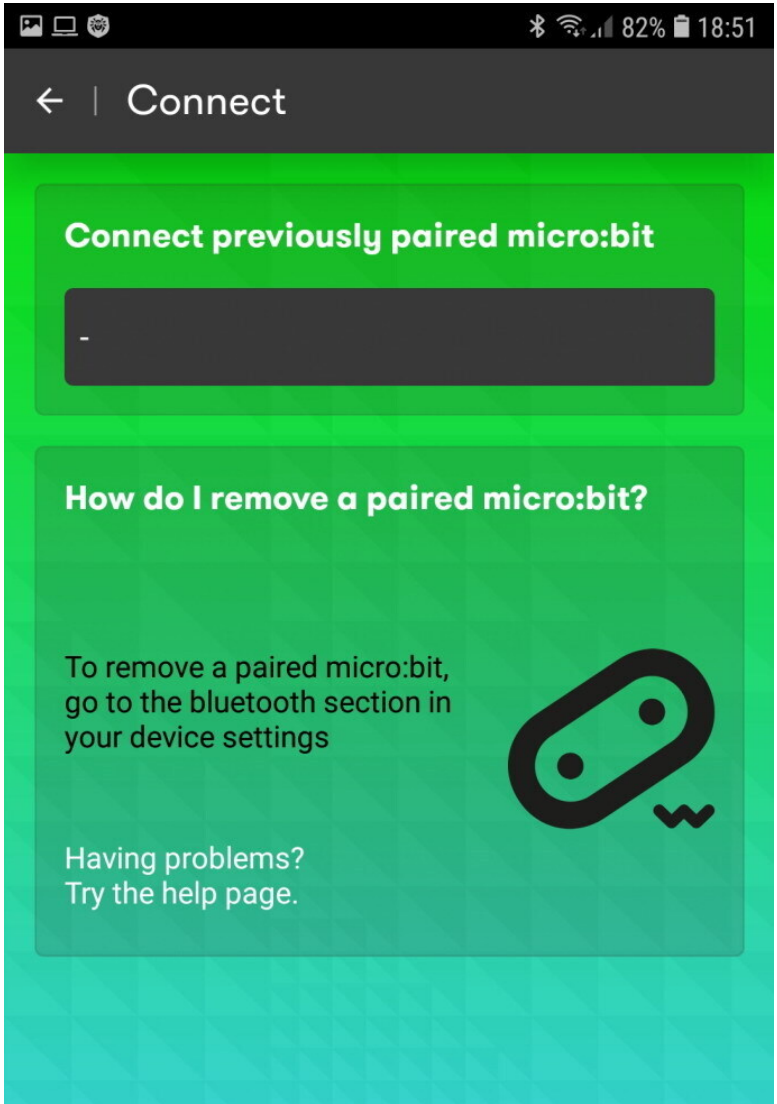
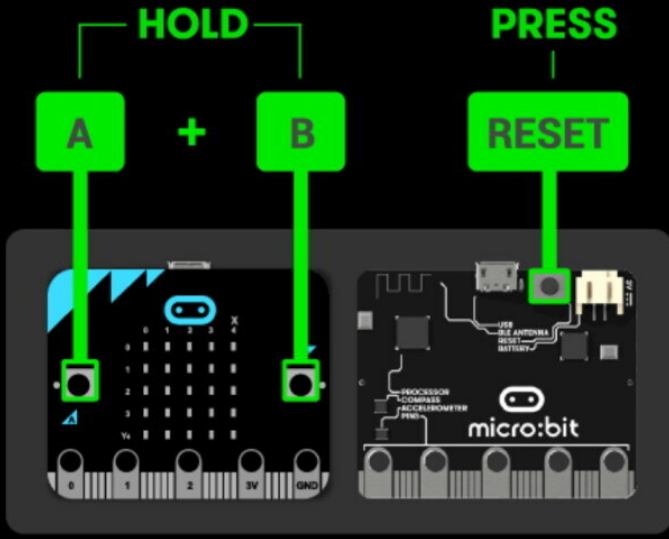

82% 18:51
Connect
Connect previously paired micro:bit
-
How do I remove a paired micro:bit?
To remove a paired micro:bit, go to the bluetooth section in your device settings
Having problems?
Try the help page.

Рис. 1.11. Подключаем micro:bit в первый раз

Теперь возьмите микрокомпьютер micro:bit в руки и нажмите одновременно кнопки A и B. Затем нажмите кнопку сброса и удерживайте некоторое время (пока на экране не высветится изображение значка Bluetooth). Далее отпустите кнопки.

В окне приложения на смартфоне вы увидите краткую инструкцию с описанием только что описанных действий (рис. 1.12).

How to pair your micro:bit



Let's do this

Step 1

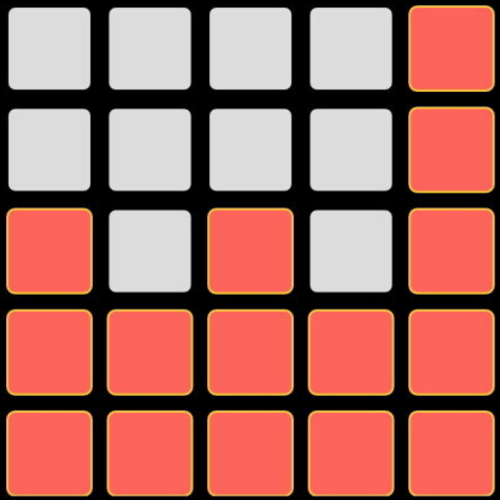
Hold the A and B buttons,

Рис. 1.12. Инструкция по подключению к micro:bit

Как только на экране устройства micro:bit высветился значок Bluetooth, нажмите в окне приложения micro:bit на смартфоне кнопку **NEXT**.

При этом на экране micro:bit появится фигурка из горящих светодиодов (образец). Вам нужно нарисовать пальцем точно такую же фигурку в окне смартфона **Enter the pattern**, а затем нажать кнопку **PAIR** (рис. 1.13).

Enter the pattern



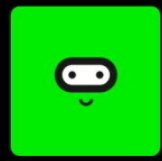
Ooh, pretty!

Step 2

Copy the pattern from your

Рис. 1.13. Образец для подключения

Для завершения процесса подключения нажмите кнопку сброса на микрокомпьютере micro:bit. Теперь связь между смартфоном (планшетом) и micro:bit установлена (рис. 1.14), и можно переходить к программированию.



You have successfully paired with micro:bit

Please press RESET button and you're done!

Рис. 1.14. Установлена связь между micro:bit и смартфоном

Учтите, что для обладателей смартфонов и планшетов с Android необходима связь с интернетом, т.к. приложение micro:bit загружает для программирования сайт https://microbit.org/code/. Здесь вам придется нажать кнопку Let's Code, после чего вы попадете на сайт https://makecode.microbit.org/, о работе с которым мы уже рассказывали.

На рис. 1.15 мы показали пример программы, созданной на этом сайте в смартфоне.

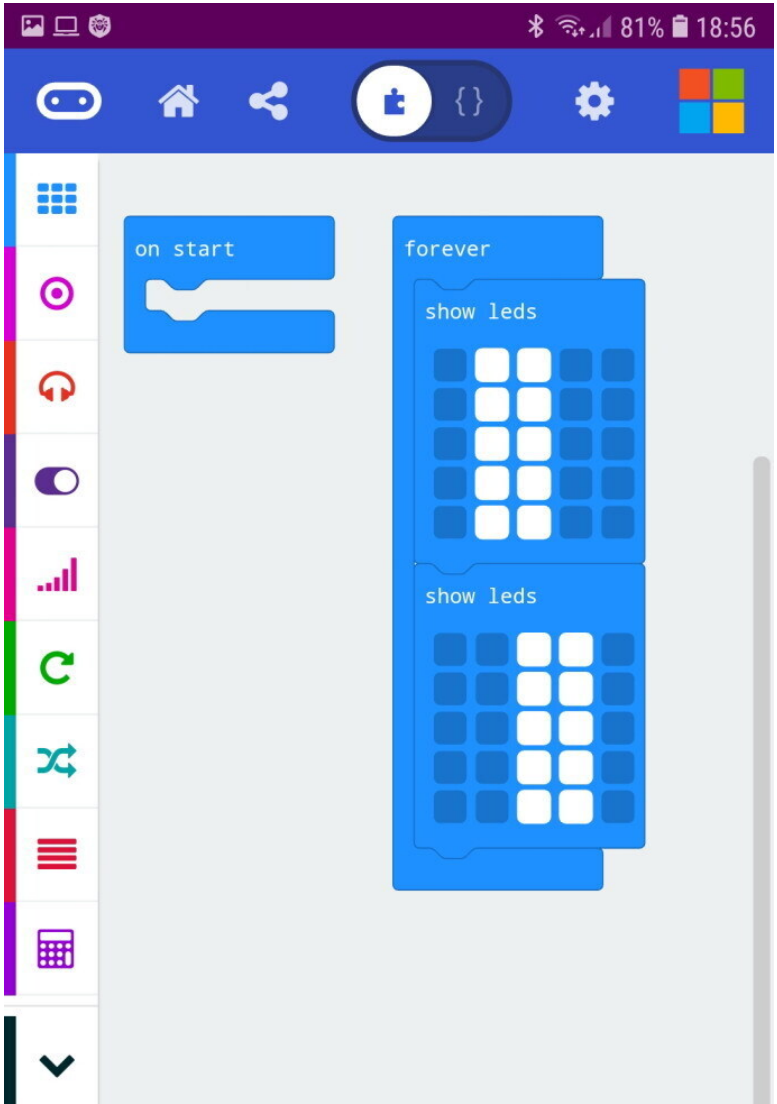

81% 18:56
{}
on start
forever
show leds
show leds

Рис. 1.15. Программа для micro:bit создана при помощи смартфона

Создав программу, загрузите ее на смартфон с помощью кнопки, расположенной слева внизу.

Далее вернитесь на главное окно приложения micro:bit и нажмите кнопку **Flash**. Выберите программу для загрузки и нажмите расположенную под ее названием кнопку **FLASH** (рис. 1.16).

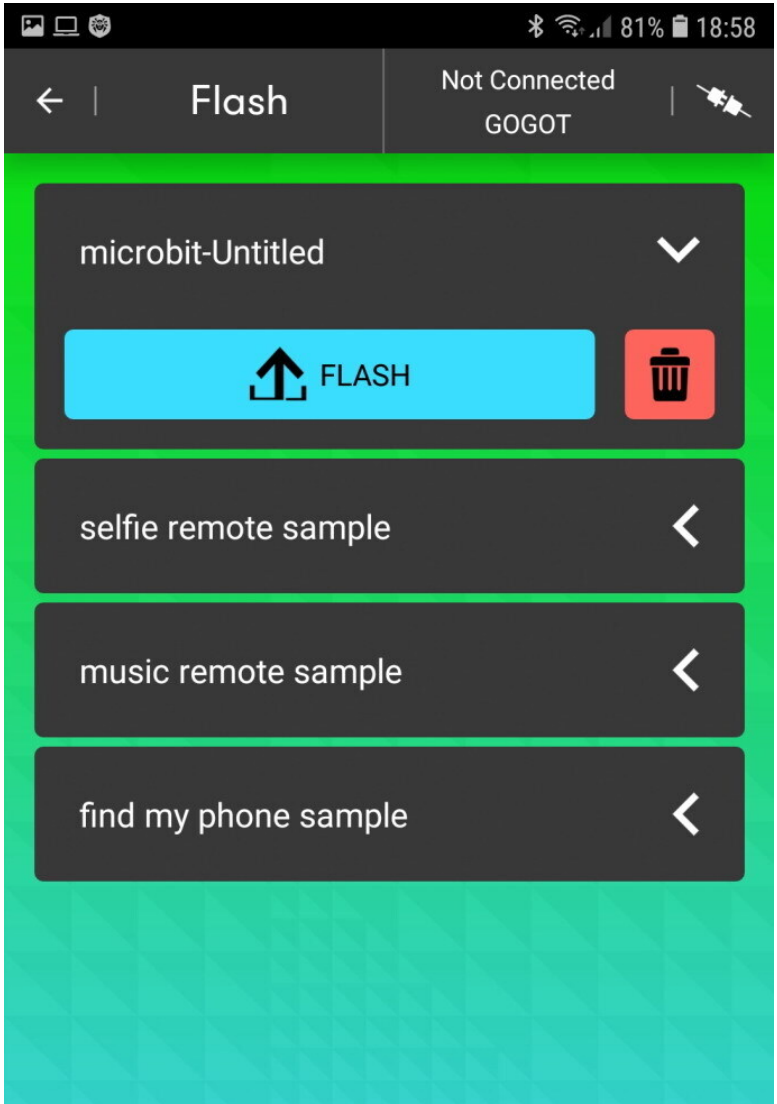

81% 18:58
Flash
Not Connected
GOGOT
microbit-Untitled
FLASH
selfie remote sample
music remote sample
find my phone sample

Рис. 1.16. Кнопка загрузки программы в micro:bit

Через некоторое время начнется загрузка программы в micro:bit (рис. 1.17). К сожалению, соединение Bluetooth довольно медленное, поэтому и загрузка будет продолжаться заметно дольше, чем через USB.

Flashing 'microbit-Untitled'

Please do not interact with micro:bit before flashing process is complete

Рис. 1.17. Процесс загрузки программы в micro:bit

После ее завершения в окне приложения появится соответствующее сообщение (рис. 1.18).

Flashing successful

You have successfully downloaded program to micro:bit

Рис. 1.18. Загрузка программы завершена

Приложение micro:bit для iPhone или iPad работает аналогично. С его помощью вы сможете программировать свой микроконтроллер и без подключения к интернету.

Мы сохранили описанную выше программу в файле **Качели**. Вы сможете найти ее в zip-архиве программ на сайте http://frolov-lib.ru/books/boxrover/, в каталоге BoxRover/ch01/microbit microbit-Качели.hex.

Итоги

В первой главе нашей книги мы подключили микроконтроллер micro:bit к компьютеру и загрузили в него самую первую программу при помощи программы Microsoft MakeCode for micro:bit.

Мы также научились создавать программы и загружать их в память micro:bit в браузере через сайт https://makecode.microbit.org/, а также через приложение micro:bit, установленное в смартфоне или планшете.

Начало положено, и теперь можно двигаться дальше!

2. Управляем светодиодами

Многие статьи и книги по программированию микроконтроллеров предлагают вам для начала написать программу, которая умеет мигать светодиодом, подключенным к микроконтроллеру через резистор. Справедливости ради нужно отметить, что на плате микроконтроллера Arduino уже есть один светодиод, которым можно мигать.

Что же касается платы micro:bit, то там есть экран из 25 светодиодов! И этими светодиодами можно не только мигать. В этой главе мы научим вас рисовать на экране различные значки, цифры и текст в режиме бегущей строки.

Что такое светодиод

Прежде чем мы приступим к созданию программ для управления светодиодами, расскажем кратко о том, что же такое светодиод. Из названия можно догадаться, что это диод, способный излучать свет.

Полупроводниковые диоды – это электронные компоненты, которые проводят ток только в одном направлении.

У диода два вывода, один из которых называется анодом, а другой – катодом. Чтобы через диод пошел ток, к аноду необходимо подключить положительный вывод батарейки, а к катоду – отрицательный. В обратную сторону ток не пойдет (на самом деле пойдет, но очень и очень маленький, он называется током утечки диода).

Светодиод излучает свет, когда через него проходит электрический ток в прямом направлении, т.е. от анода к катоду.

Никогда не подключайте диоды и светодиоды к батарейке напрямую – через диод может пойти слишком большой ток и он в итоге выйдет из строя. Обязательно используйте токоограничительный резистор.

В продаже вы можете встретить светодиоды, допускающие прямое подключение к батарейке напряжением до 5 В без токоограничительного резистора, однако это нужно уточнить у продавца.

На рис. 2.1 мы показали, как можно подключить светоди-

од к батарейке с напряжением 1,5 В.

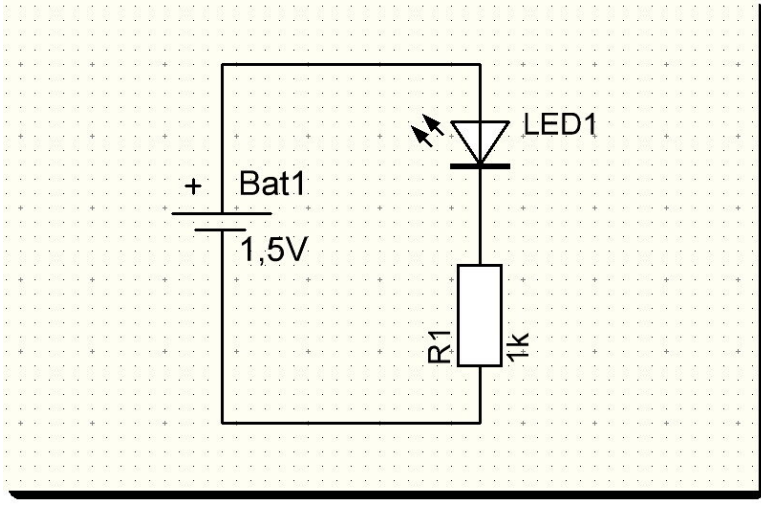


Рис. 2.1. Подключение светодиода к батарейке

При использовании батарейки с напряжением 1,5-3 В обычный светодиод нужно подключать через резистор номиналом 1 К. Этот резистор ограничивает ток, проходящий через светодиод.

Собирая такую схему самостоятельно, убедитесь, что свет излучается только при правильной полярности, т.е. когда ток через светодиод идет в прямом направлении, т.е. от анода к катоду.

Возможно, это будет для вас сюрпризом, но носители

электрического тока, а именно электроны, перемещаются в обратном направлении, от минуса к плюсу – они несут отрицательный электрический заряд. До открытия электрона Томсоном в 1897 году природа электрического тока еще не была до конца изучена, поэтому было принято условное направление движения тока – от плюса к минусу. Так оно осталось и до сих пор.

Создаем программу для мигания светодиодом

Откройте программу MakeCode (установленную на компьютер или загруженную в браузер). Создайте там новый проект, как это мы описали в предыдущем разделе книги, и раскройте палитру **Светодиоды** (рис. 2.2).

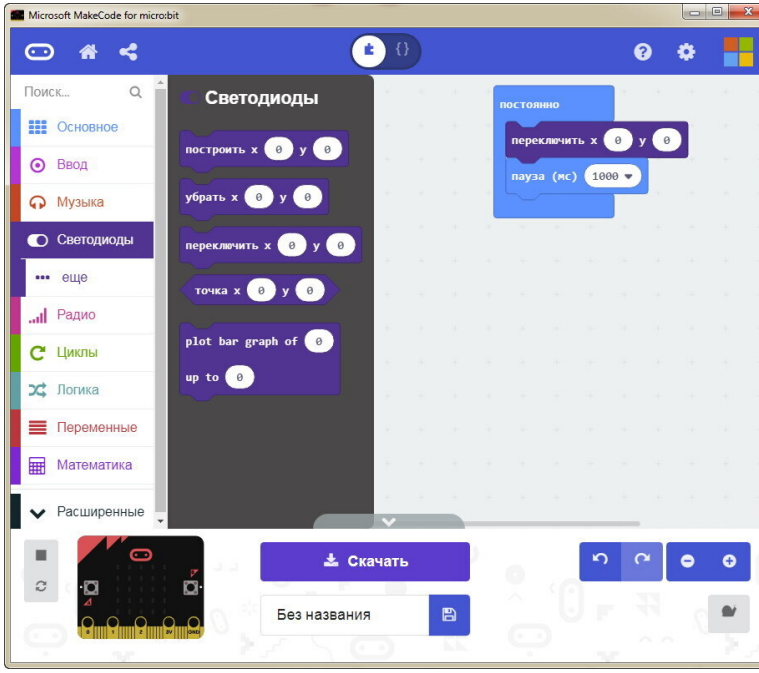


Рис. 2.2. Палитра Светодиоды

Перетащите блок **переключить** в блок **постоянно**. После этого откройте палитру **Основное** и перетащите из нее блок **пауза**, расположив его под блоком **переключить**, как это показано на рис. 2.2.

Теперь щелкните кнопку **Скачать**, и через некоторое время вы сможете наблюдать, как на плате micro:bit начнет мигать светодиод с координатами (0, 0). Интервал мигания составит, как и следовало ожидать, 1 сек.

На рис. 2.3 показан момент, когда светодиод включен.

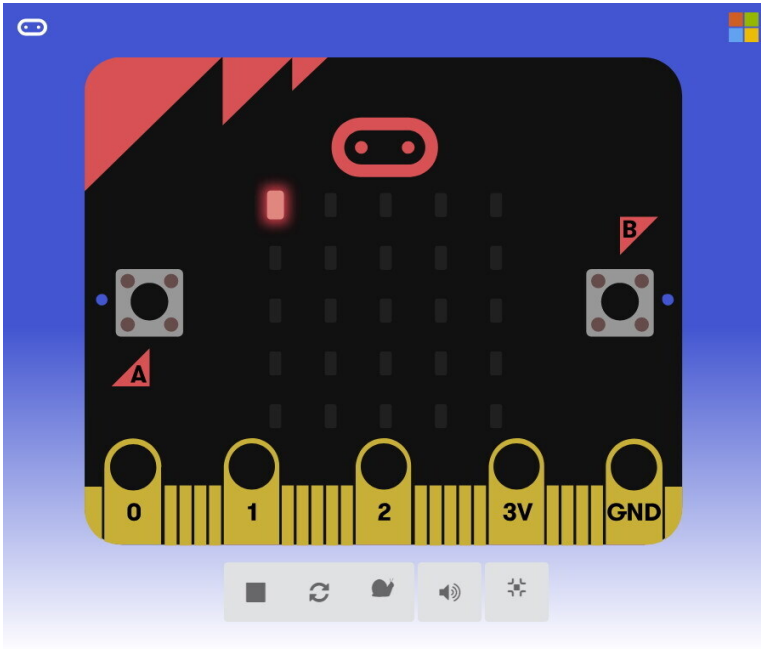


Рис. 2.3. Мигает светодиод с координатами (0,0)

Система координат для выбора светодиодов показана на рис. 2.4.

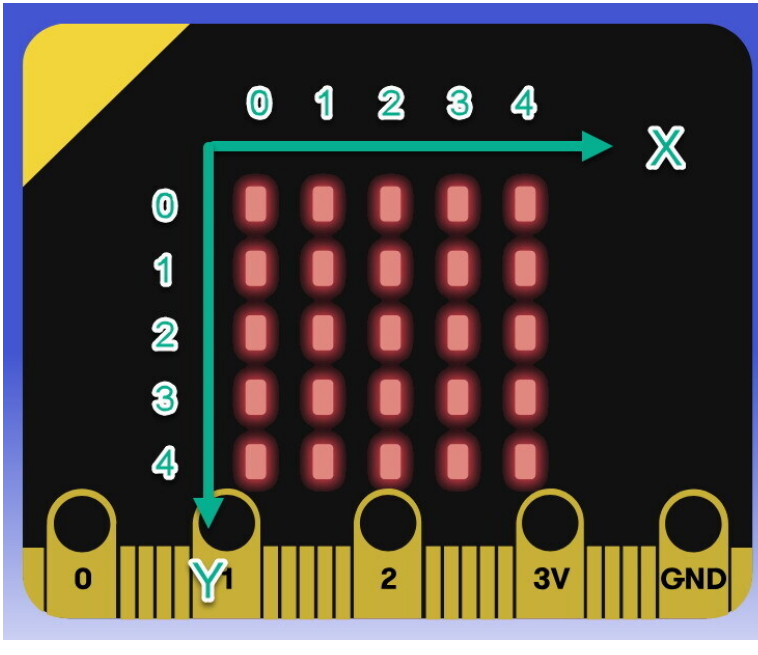


Рис. 2.4. Система координат для выбора светодиодов

Если расположить микроконтроллер разъемом вниз, то начало системы координат будет в левом верхнем углу. Ось X пойдет вправо, а ось Y – вниз.

Все, что вы добавите в блок **постоянно**, будет выполняться в зацикленном режиме, как бы по кругу. Вначале сработает блок **переключить**, потом **пауза**, дальше опять **переключить** и **пауза**, и так бесконечно, пока на микроконтрол-

лер подается напряжение питания.

В палитре **Основное** также есть блок **при начале**, содержимое которого исполняется только один раз после включения питания микроконтроллера.

Теперь давайте сохраним проект с помощью кнопки с изображением дискеты, задав для него имя «Мигаем-светодиодом». Код программы будет сохранен в файле BoxRover/ch02/microbit-Мигаем-светодиодом.hex, и вы сможете его скачать на сайте автора http://frolov-lib.ru/books/boxrover/.

Теперь давайте проведем некоторые эксперименты.

Прежде всего, попробуйте изменить расположение мигающего светодиода, отредактировав его координаты в блоке **переключить**.

Если щелкнуть мышью значение координаты, появится слайдер, с помощью которого можно выбрать число от 0 до 4. Число также можно задать и с клавиатуры, если предварительно щелкнуть мышью соответствующее поле. Установите координаты светодиода (4,4), как это показано на рис. 2.5.

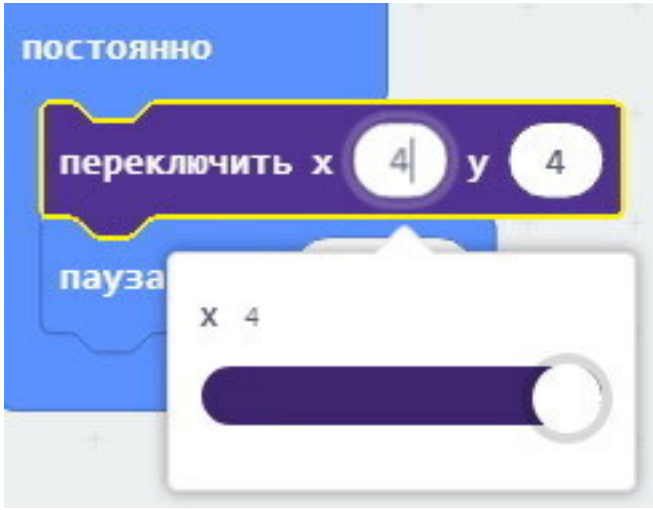


Рис. 2.5. Изменяем координаты мигающего светодиода

Если теперь загрузить программу в память micro:bit, то будет мигать уже другой светодиод (рис. 2.6).

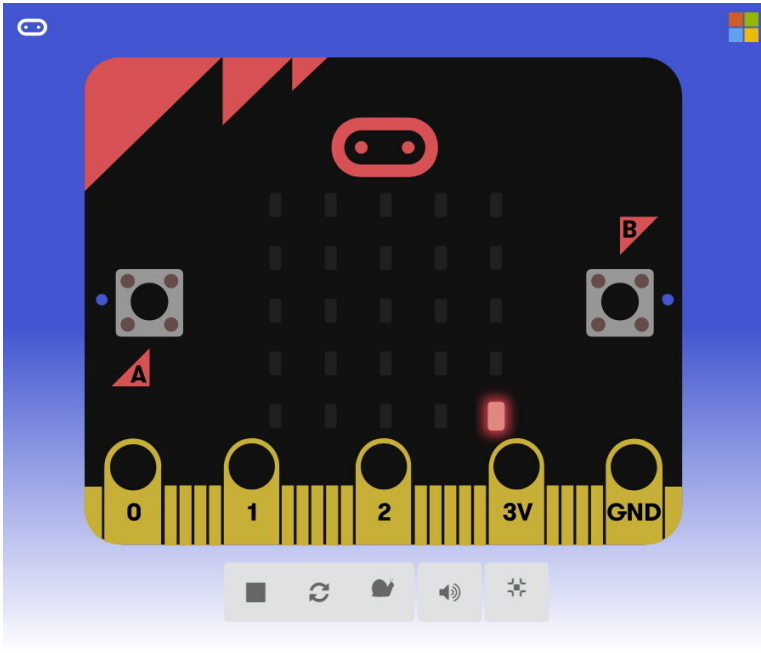


Рис. 2.6. Мигает светодиод с координатами (4,4)

В качестве следующего эксперимента попробуйте уменьшить интервал мигания, изменив значение в панели **пауза**.

Установите там, например, задержку 100 мс. Мигание светодиода заметно ускорится. Вы даже можете совсем убрать задержку из блока постоянно (рис. 2.7) .

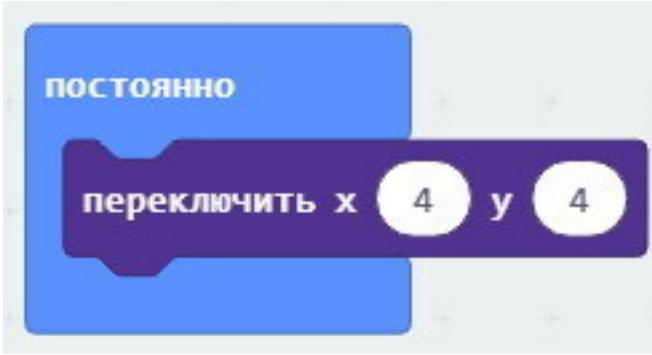


Рис. 2.7. Цикл переключения светодиода без задержки

Светодиод начнет мигать еще быстрее.

Рисуем линию

Итак, мы только что научились мигать любым из 25 светодиодов, имеющихся в нашем распоряжении на плате микроконтроллера micro:bit. Теперь поставим перед собой более сложную задачу – нарисуем на светодиодном дисплее линию. Для этого нужно включить несколько светодиодов, расположенных вдоль линии.

Для того чтобы включить светодиод, можно воспользоваться блоком **построить** из панели компонентов **Светодиоды**. Этому блоку нужно передать в качестве параметров два значения – координаты светодиода, который нужно включить.

Проще всего добавить в блок **при начале** несколько блоков **построить**, задав для каждого из них нужные координаты.

Вы можете перетащить сначала один такой блок, а потом просто продублировать его, щелкнув правой клавишей мыши и выбрав из появившегося контекстного меню строку **Дублировать**.

На рис. 2.8 мы добавили пять блоков **построить**, указав для них координату по оси X, равную 0. По оси Y мы задали значения от 0 до 4.

Рис. 2.8. Включаем пять светодиодов вдоль линии

Если загрузить такую программу, то мы увидим, что на дисплее нашего микроконтроллера появилась вертикальная

линия (рис. 2.9).

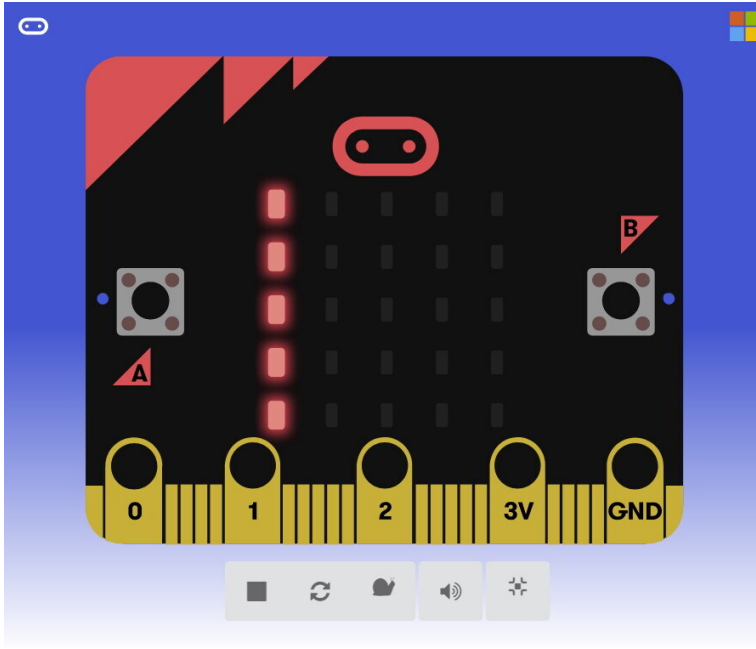


Рис. 2.9. Мы нарисовали линию

Код этой программы мы сохранили в файле BoxRover/ch02/microbit-Линия1.hex.

Но повторять одни и те же блоки – скучное дело. Давайте нарисуем линию с помощью так называемого цикла.

Что такое цикл?

В программировании цикл представляет собой конструк-

цию, позволяющую организовать многократное выполнение одних и тех же операций.

Сразу перейдем от слов к делу.

Добавьте в блок **при начале** из палитры **Циклы** блок **для**, а внутрь него поместите блок построить, как это показано на рис. 2.10.

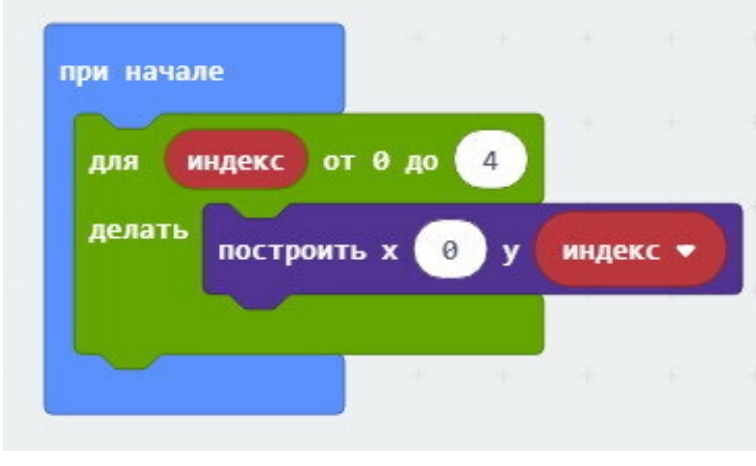


Рис. 2.10. Рисование линии с помощью цикла

В блоке **для** имеется так называемая переменная цикла **индекс**. Задайте для этой переменной конечное значение в поле **до**, равное 4 (мышью с помощью слайдера или с клавиатуры).

Перетащите мышью переменную **индекс** из поля **для** в

поле **y** блока **построить**.

Тело нашего цикла (т.е. блок **построить**) выполнятся пять раз, при этом переменная **индекс** будет принимать последовательно значения от 0 до 4.

В результате на дисплее нашего micro:bit будет нарисована точно такая же вертикальная линия, что показана выше на рис. 2.9. Но теперь в программе нет скучного повторения блоков.

Теперь нарисуем линию по диагонали монитора micro:bit. Для этого измените программу так, как это показано на рис. 2.11.

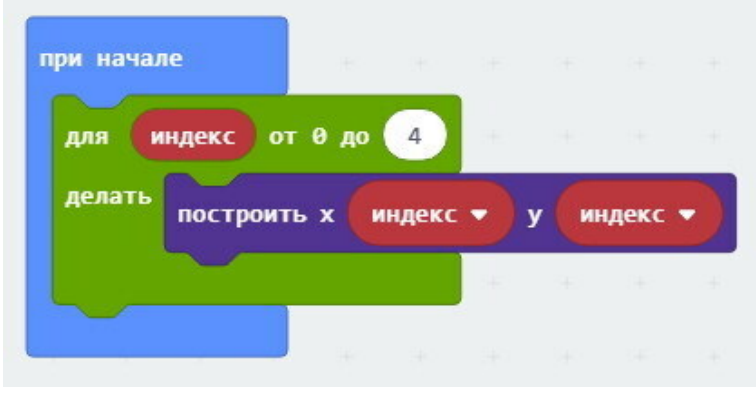


Рис. 2.11. Рисование линии по диагонали

Теперь в цикле одновременно будут изменяться координаты по обеим осям, X и Y. Результат показан на рис. 2.12.

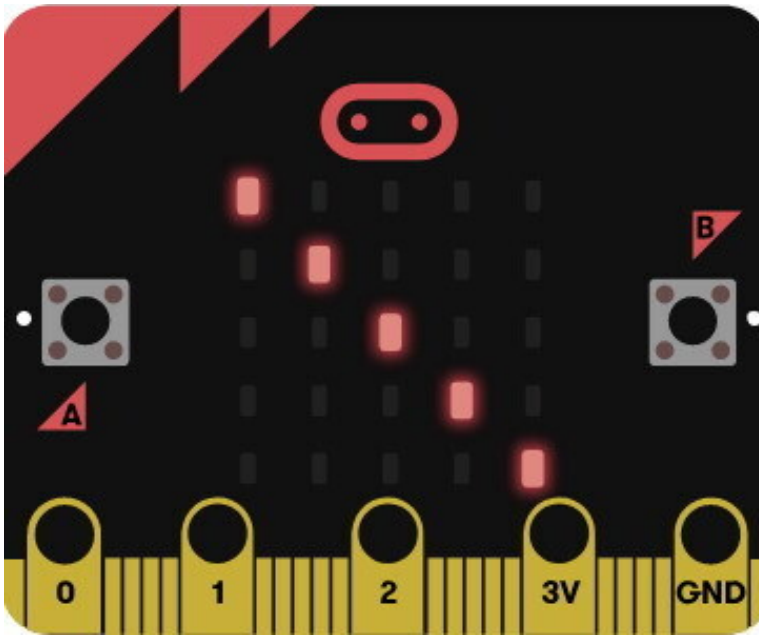


Рис. 2.12. Наклонная линия на мониторе micro:bit

Код программы, рисующий наклонную линию, мы сохранили в файле BoxRover/ch02/microbit-Линия.hex.

Рисуем на экране micro:bit фигуры и значки

Как мы уже говорили, на экране платы micro:bit есть 25 светодиодов, что позволяет рисовать простейшие фигуры. Но делать это, зажигая каждый нужный светодиод отдельно – довольно утомительная задача. К счастью, в редакторе MakeCode есть удобные средства для рисования на экране различных значков.

Откройте палитру **Основное** и добавьте в блок **при начале** блок **показать значок** (рис. 2.13).

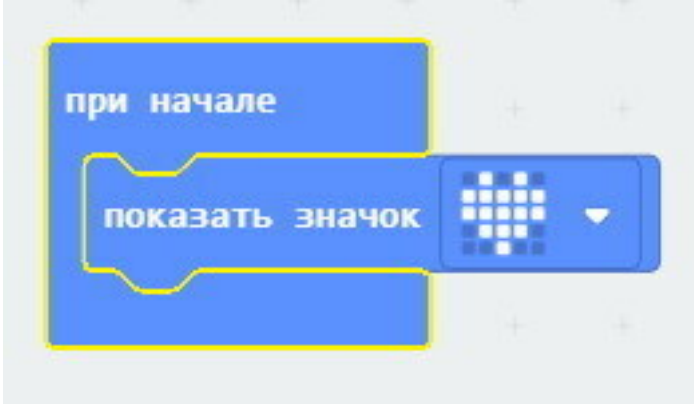


Рис. 2.13. Рисуем значок

По умолчанию будет нарисовано сердечко, но вы можете задать и другие значки. Чтобы увидеть их список, щелкните изображение треугольника в блоке **показать значок**. На экране появится палитра доступных значков (рис. 2.14).

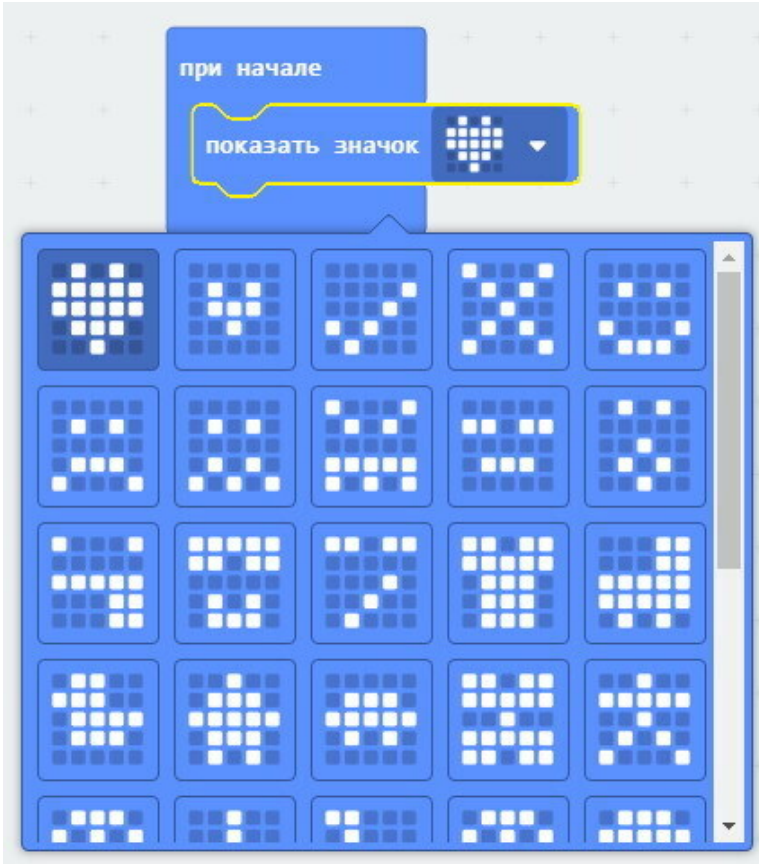


Рис. 2.14. Палитра доступных значков

Как видите, значков довольно много.

Если загрузить нашу программу в память микроконтрол-

лера, на его мониторе появится изображение выбранного вами значка (рис. 2.15).

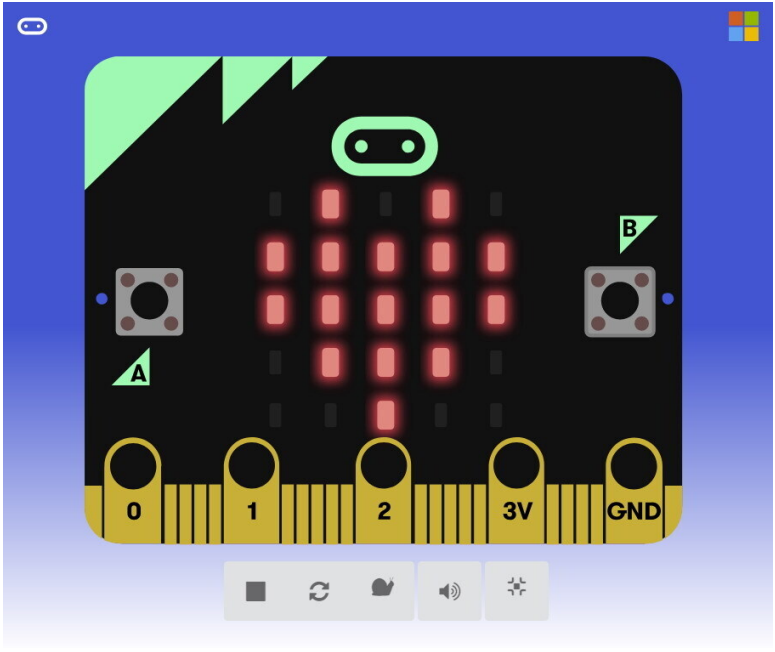


Рис. 2.15. Изображение значка на мониторе микроконтроллера

Давайте теперь составим программу, которая будет показывать по очереди разные значки.

Прежде всего, добавьте в блок постоянно отображение большого и маленького значка сердца, как это показано на

рис. 2.16.

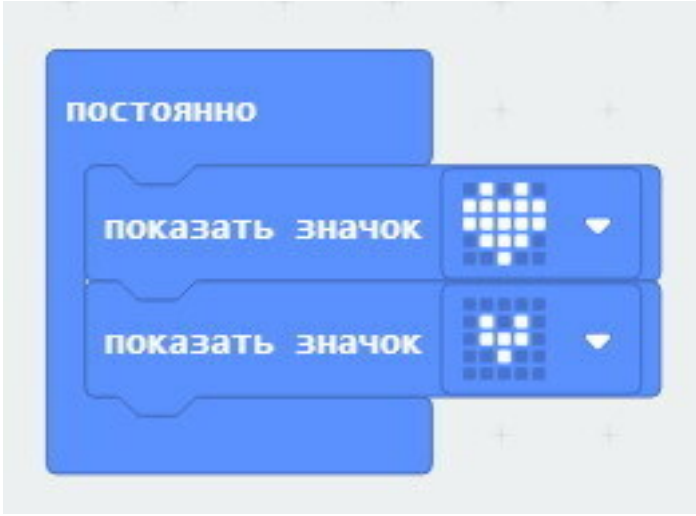


Рис. 2.16. Значки будут показаны по очереди

Эти значки будут показываться в бесконечном цикле по очереди, и вы увидите на мониторе, как бьется сердце!

Код программы записан в файл BoxRover/ch02/microbit-Сердце.hex.

Вы можете показывать несколько значков по очереди, включив их в цикл (рис. 2.17) .

Рис. 2.17. Показ нескольких значков в цикле

Здесь мы не используем задержку, но вы можете попробовать добавить между блоками **показать значок** блок **пауза**. Это позволит регулировать время отображения значков на мониторе.

Но что, если вам нужно показать значок, которого нет в палитре блока **показать значок**?

Ничего страшного!

Просто воспользуйтесь блоком **показать светодиоды**. В этом блоке вы можете нарисовать мышью любой значок, который вам потребуется.

Если добавить нарисованные таким образом значки в блок **постоянно**, на мониторе появится созданная вами анимация (рис. 2.18).


постоянно
показать светодиоды
показать светодиоды

Рис. 2.18. Рисуем в блоке показать светодиоды свои значки

Код программы вы найдете в файле microbit-Анимация.hex.

Рисуем стрелки

Специально для того чтобы показывать на мониторе стрелки, предусмотрен блок **показать стрелку с направлением**.

Добавьте в блок **постоянно** восемь блоков показать стрелку направлением, задав для каждой свое направление. Если сделать это так, как показано на рис. 2.19, то после запуска программы на мониторе появится вращающаяся стрелка (рис. 2.20).

Рис. 2.19. Программа для рисования вращающейся стрелки

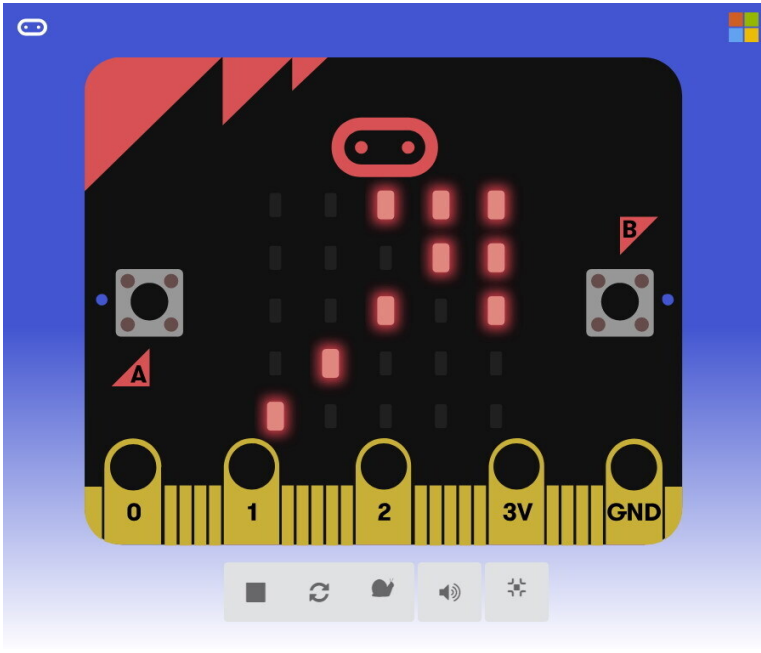


Рис. 2.20. Изображение вращающейся стрелки на мониторе микрокомпьютера

Программа вращающейся стрелки есть в архиве под именем microbit-Стрелка.hex.

Выводим на монитор число

В палитре **Основное** есть блок **показать число**, с помощью которого вы можете выводить на монитор micro:bit произвольные числа.

Ниже на рис. 2.21 мы показали программу, которая показывает в цикле числа от 0 до 15.

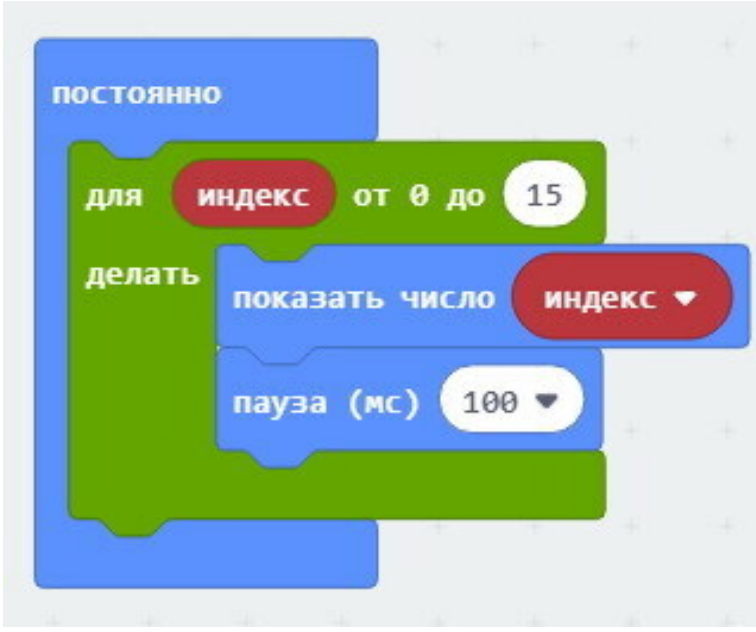


Рис. 2.21. Программа для отображения чисел от 0 до 15 на мониторе микрокомпьютера

Здесь мы добавили из палитры **Циклы** блок **для**. Он позволяет организовать исполнение своего тела заданное количество раз, при этом переменная цикла (в нашем случае это переменная **индекс**) будет изменять свое значение в заданных пределах с шагом 1.

В нашей программе цикл отработает 16 раз, при этом вы увидите на мониторе micro:bit числа от 0 до 15 (рис. 2.22).

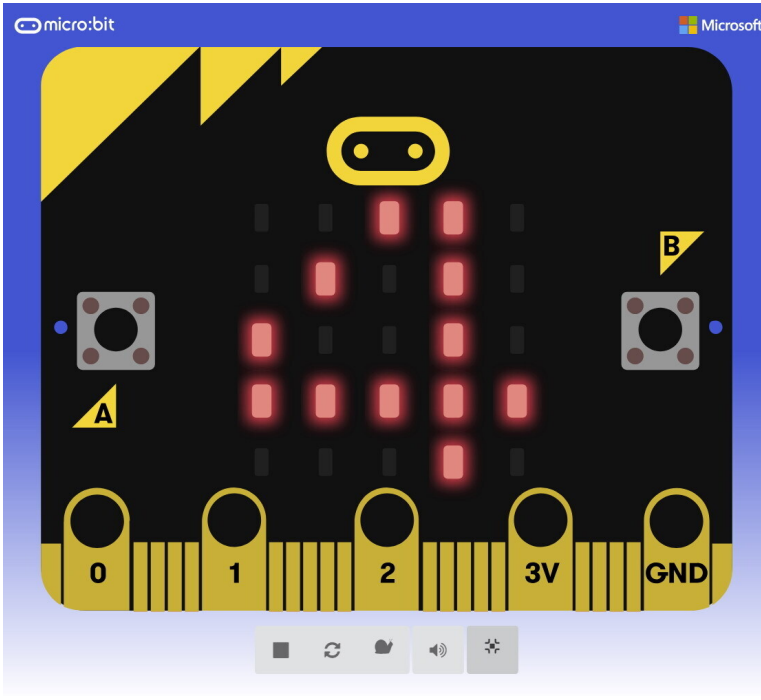


Рис. 2.22. Показываем числа на мониторе micro:bit

Обратите внимание, что числа, большие девяти, показываются в режиме бегущей строки, так как иначе они бы не поместились в матрице светодиодов.

Программа сохранена в файле microbit-Числа.hex.

Выводим на монитор бегущую текстовую строку

А что если попытаться показать на мониторе микрокомпьютера текст? Так как светодиодов маловато, то текст будет выведен в виде бегущей строки. Но и этого в некоторых случаях может оказаться достаточно, например, для показа температуры.

Добавьте в блок при начале блок показать строку из палитры **Основное** (рис. 2.23).

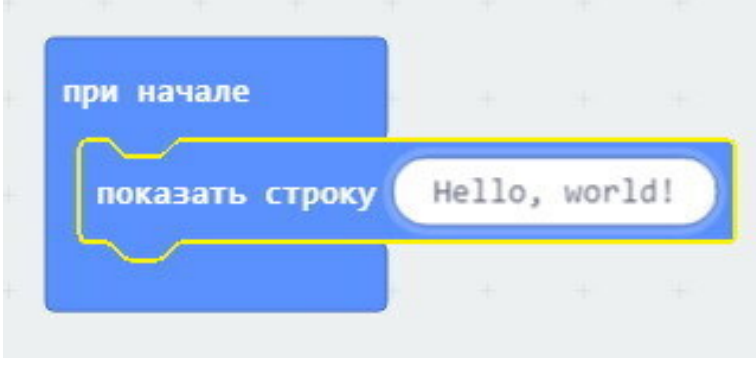


Рис. 2.23. Программа выводит на монитор бегущую строку

Программа есть в архиве под именем microbit-Hello-

world.hex. После ее запуска на мониторе появится бегущая строка «Hello, world!» (рис. 2.24).

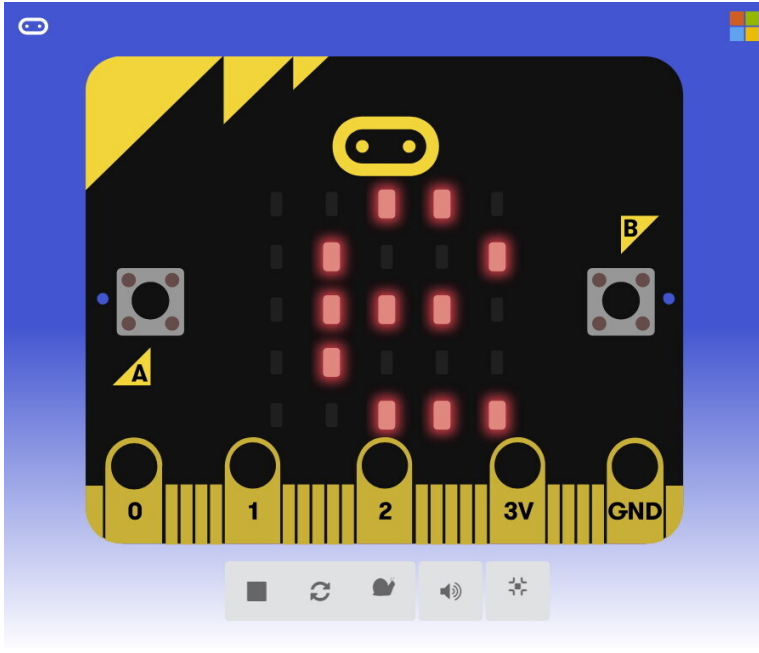


Рис. 2.24. Бегущая строка на мониторе вашего micro:bit

К сожалению, в строке можно использовать только латинские символы, цифры и знаки – символы кириллицы показать не получится.

Управляем яркостью светодиодов

В палитре **Светодиоды … еще** есть блоки, предоставляющие дополнительные возможности управления светодиодами. В частности, с помощью некоторых из них можно управлять яркостью светодиодов (рис. 2.25).

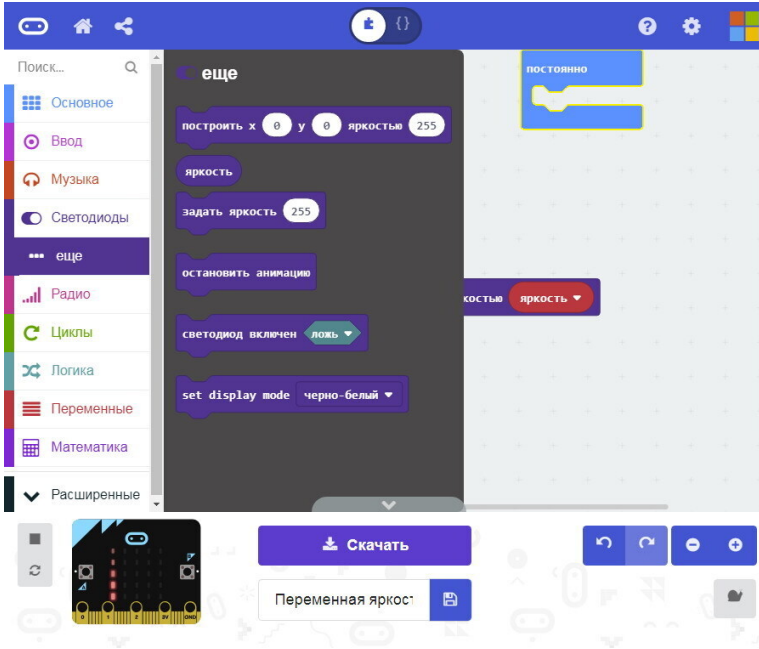


Рис. 2.25. Блоки с дополнительными возможностями управления светодиодами

Прежде всего, раскройте палитру **Переменные**, и с помощью кнопки **Создать переменную** добавьте переменную с именем **яркость**. Просто введите имя создаваемой переменной в окне, которое откроется после щелчка этой кнопки.

Созданная вами переменная появится в палитре **Переменные** (рис. 2.26).

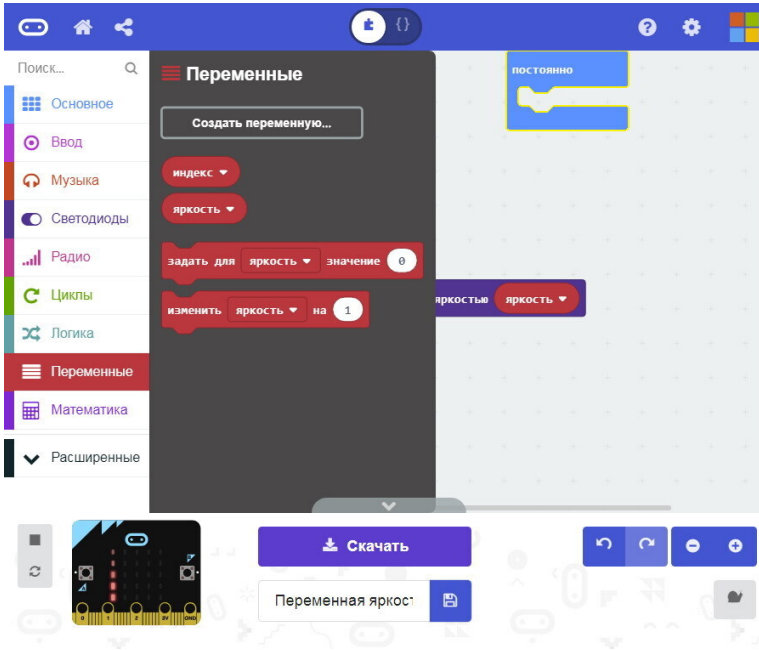


Рис. 2.26. Мы создали переменную с именем яркость

Но что такое переменная?

Можно считать, что созданная вами переменная – это имя места в памяти микроконтроллера, где будет хранится цифровое значение. В нашем случае это будет значение яркости светодиода, которое может изменяться в интервале от 0 до 255.

При необходимости вы можете создать несколько переменных для хранения различных значений, задав для каждой такой переменной свое собственное имя.

Теперь подготовьте программу, показанную на рис. 2.27. Эта программа нарисует вертикальную линию из пяти светодиодов, причем у всех светодиодов будет разное значение яркости.

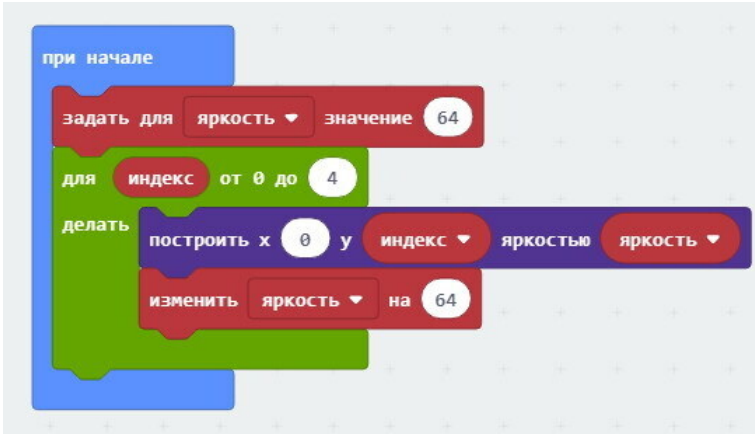


Рис. 2.27. Программа для управления яркостью светодиодов

Результат работы программы, сохраненной в файле microbit-Переменная-яркость-линии.hex, вы можете увидеть на рис. 2.28.

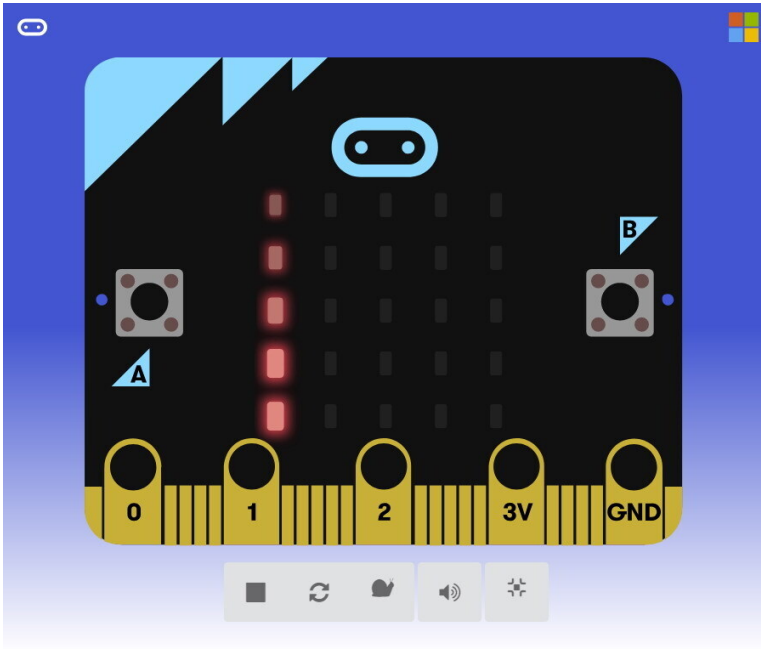


Рис. 2.28. Линия из светодиодов с различной яркостью

Как работает наша программа?

Прежде всего, для переменной **яркость** мы задаем начальное значение 64. Далее запускается цикл **для**, параметр **индекс** которого изменяет свое значение от 0 до 4, как и в предыдущей программе.

Однако теперь для включения светодиода мы используем блок **построить яркостью**. Этот блок зажигает светодиод с

координатами (x, y) и дополнительно устанавливает для этого светодиода яркость, указанную в последнем параметре.

После зажигания светодиода в дело вступает блок **изменить**, который увеличивает значение нашей переменной яркость на 64 (чтобы уменьшать значения, задайте отрицательное число).

Теперь при следующем проходе цикла яркость зажигаемого светодиода увеличится на значение 64.

Доверимся случаю

В палитре **Математика** есть интересный блок **выбрать случайно**. Он позволяет получить случайное число в заданном вами диапазоне. Давайте используем случайные числа при выключении светодиодов.

На рис. 2.29 мы привели пример такой программы, в которой все случайно!

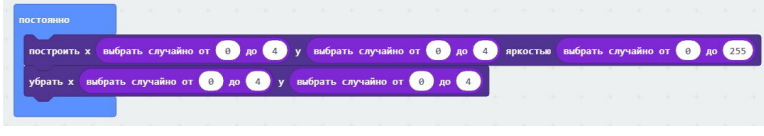


Рис. 2.29. Программы случайно зажигает светодиоды со случайной яркостью

В бесконечном цикле программа с помощью блока **построить** зажигает светодиоды, при этом координаты (x,y) светодиода, а также его яркость, задается случайно.

Как видите, координаты выбираются случайно в диапазоне от 0 до 4, а яркость – в диапазоне от 0 до 255.

После включения случайного светодиода программа с помощью блока **убрать** выключает светодиод, координаты которого также выбираются случайно (рис. 2.30).

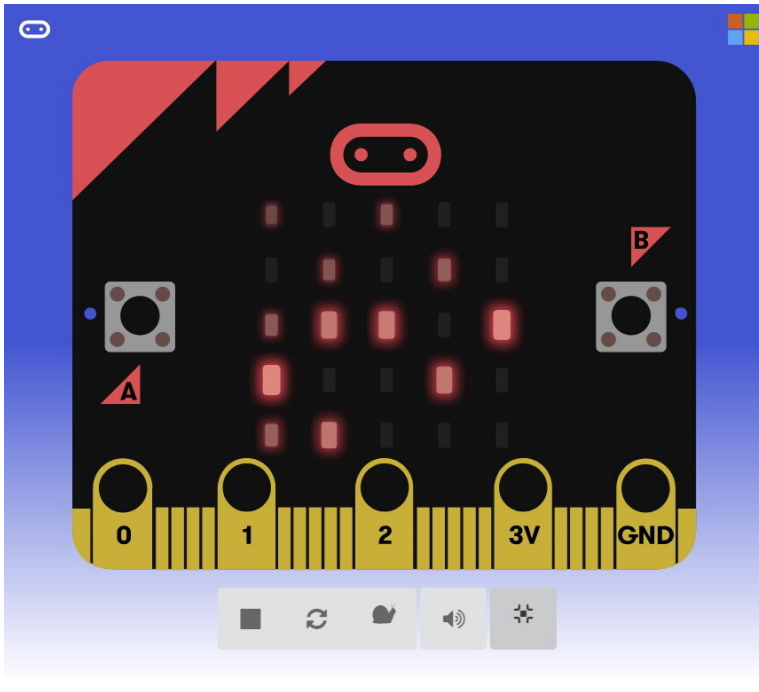


Рис. 2.30. Светодиоды зажигаются и гаснут случайно

Программа звездного неба сохранена в файле microbit-Случайно.hex.

Возможно, кто-то найдет, что это все похоже на звездное небо, где звезды зажигаются, живут и гаснут через какое-то время. Если вам кажется, что звезды гаснут слишком быстро, добавьте небольшую задержку (например, 100 мс) в ко-

нец цикла (рис. 2.31).

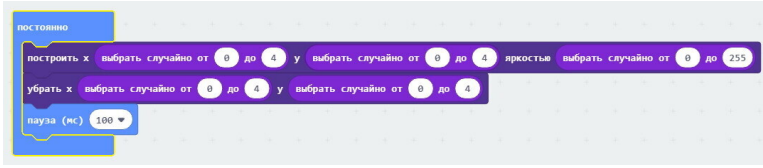


Рис. 2.31. Задержка замедляет процесс управления светодиодами

А теперь давайте составим сложную программу, которая будет перемещать горящую точку по экрану микроконтроллера случайным образом.

Готовая программа показана на рис. 2.32, вы можете загрузить ее из файла microbit-Блуждающая-точка.hex. Рассмотрим подробно, что она делает.

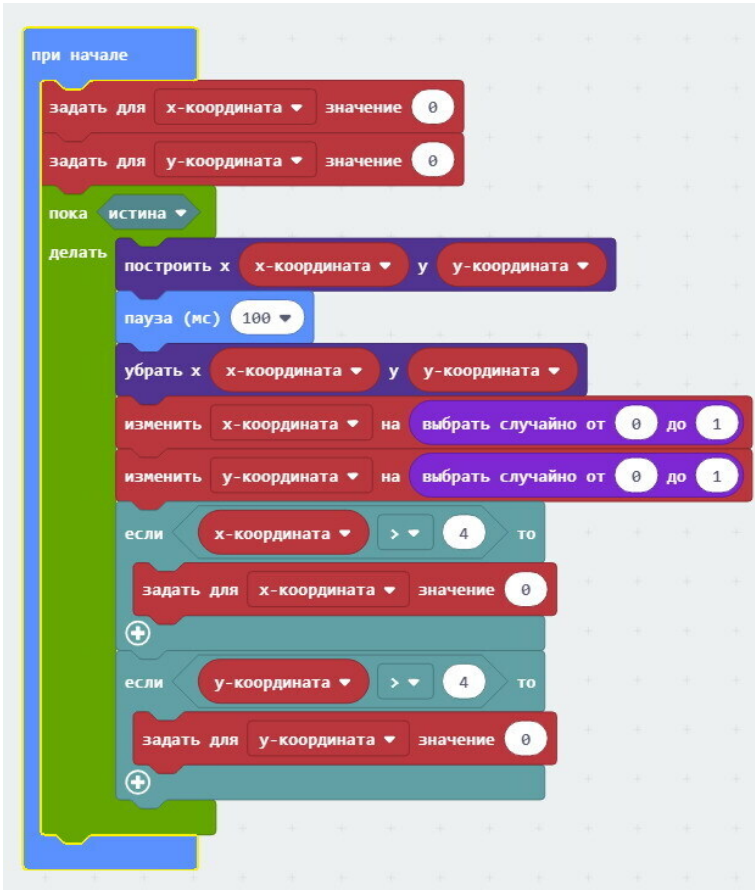


Рис. 2.32. Рисуем на экране случайно блуждающую точку

Как видите, программа запускается один раз в блоке **при**

начале.

В палитре **Переменные** мы определили две переменные с именами **х-координата** и **у-координата**. В первой из них мы будем хранить текущую координату нашей блуждающей точки по оси X, а во второй – по оси Y. Сразу после запуска мы инициализируем эти переменные нулевыми значениями, поэтому свое путешествие точка начнет из верхнего левого угла экрана micro:bit.

После инициализации переменных запускается бесконечный цикл **пока**. Мы добавили его из палитры **Циклы**.

У цикла **пока** есть параметр, который анализируется перед каждым очередным проходом цикла. По умолчанию этот параметр имеет значение **истина**, поэтому наш цикл никогда не завершится. Вы можете задать в этом параметре какое-либо условие, которое ограничит блуждание нашей точки, однако сейчас мы этого делать не будем.

Что же внутри цикла?

Прежде всего мы зажигаем светодиод в блоке **построить** в точке с координатами (х-координата, у-координата), ждем 100 мс при помощи блока **пауза**, а затем гасим светодиод в блоке **убрать**.

Дальше мы изменяем переменные **х-координата** и **у-координата** случайным образом, при этом изменение выбирается в диапазона от 0 до 1. Это означает, что координаты по осям X и Y могут либо не измениться вовсе, либо увеличатся на единицу.

После изменения координат нам нужно проверить, что координаты не вышли за границу нашего монитора – ведь у нас всего пять рядов по пять светодиодов в каждом. Для проверки мы используем блоки **если**, которые нужно добавить из палитры **Логика**.

Тело блока **если** выполняется в том случае, когда верно условие, добавленное в качестве параметра блока. Добавьте из палитры **Логика** условие сравнения числовых операторов, как это показано на рис. 2.32.

Если значение новой координаты по оси X или Y получилась больше 4, то мы задаем для соответствующей переменной нулевое значение. Тут потребуется блок **задать значение** из палитры **Переменные**.

Как работает наша программа?

Когда в результате случайных изменений новые координаты блуждающей точки не выходят за пределы экрана, то точка передвигается (или нет) на один шаг, либо по одной оси, либо по обеим осям координат сразу.

Но если новая позиция выходит за пределы возможного, то ее новая координата устанавливается равной нулю. Визуально точка перепрыгивает в начало соответствующей оси координат. Если же в результате случая координаты по обеим осям становятся больше 4, то точка прыгает в левый верхний угол, с координатами (0,0).

Изображение блуждающей точки показано на рис. 2.33.

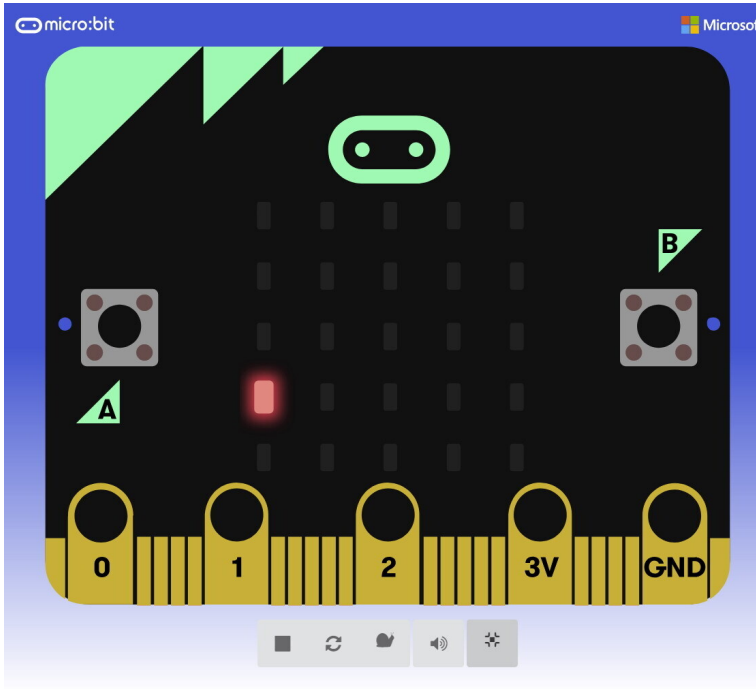


Рис. 2.33. Светящаяся точка блуждает случайным образом по монитору микрокомпьютера

Домашнее задание

Если текст или число не помещается целиком на экран micro:bit, то используется вывод в режиме бегущей строки. Но что если нам нужно вывести на экран графическое изображение, превышающее размер монитора?

Оказывается, и это возможно.

В качестве домашнего задания попробуйте написать программу, которая будет рисовать на экране micro:bit бегущую волну.

Подсказка: используйте палитру **Изображения**. Там вам будет нужен блок **создать большое изображение** и **scroll image with offset and interval** (сдвиг изображения со смещением и интервалом).

Решение вы сможете найти в файле BoxRover/ch02/microbit-Волна.hex из архива, который можно скачать на сайте автора этой книги http://frolov-lib.ru/books/boxrover/. Прежде чем скачивать этот файл, попробуйте создать программу самостоятельно!

Итоги

Итак, во второй главе книги мы научились выводить различную информацию на экран micro:bit, состоящего из 25 светодиодов.

Теперь мы умеем зажигать и гасить светодиоды в нужном нам месте, и задавать их яркость. Мы можем рисовать на экране различные фигуры, стрелки, и даже выводить бегущую текстовую строку.

Мы познакомились с генератором случайных чисел и сделали с его помощью анимированное изображение крохотного звездного неба, а также анимацию случайно блуждающей точки.

Надеюсь, вам также удалось сделать свое первое домашнее задание – нарисовать на экране micro:bit бегущую волну.

В следующей главе мы продолжим изучение оборудования, которое есть на плате micro:bit. Вы научитесь пользоваться кнопками для изменения текущего хода работающей программы и выполнения других действий.

3. Работаем с кнопками

Обычная кнопка представляет собой аппаратное устройство, предназначенное для замыкания и размыкания электрической цепи. Когда кнопка нажата, цепь замкнута и по ней идет электрический ток, а когда отжата – цепь разомкнута, ток не идет.

Также вам наверняка знакомы так называемые сенсорные кнопки. Чтобы такая кнопка сработала, достаточно до нее дотронуться.

На рис. 3.1 мы добавили обычную кнопку SA1 в схему подключения светодиода к батарейке. Если нажать кнопку, светодиод загорится, если отпустить – погаснет.

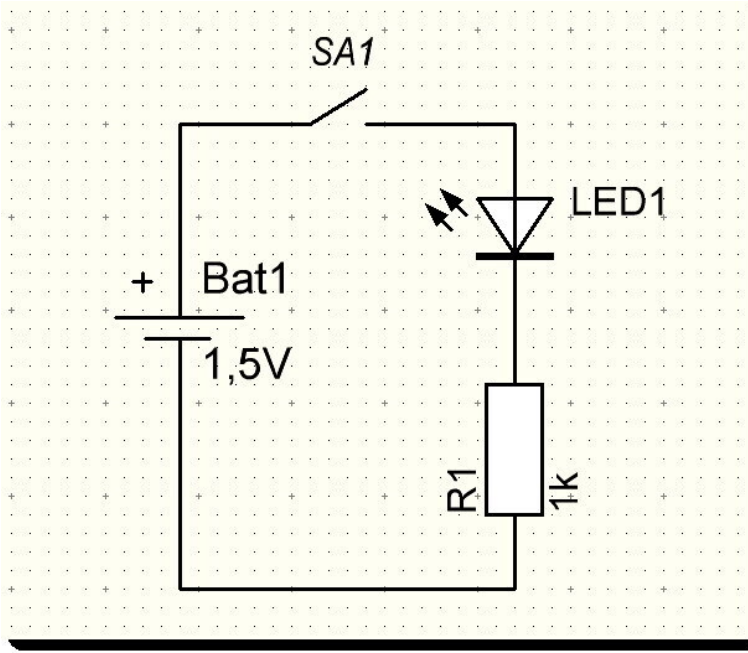


Рис. 3.1. Схема для включения светодиода с помощью кнопки

Различного рода кнопки есть практически везде – в пультах управления телевизором, в мобильных телефонах, в клавиатуре компьютера и во многих других устройствах. Некоторые из них обычные, а некоторые – сенсорные.

В микрокомпьютере micro:bit версии 1.5 есть три кнопки. Одна из них расположена с обратной стороны устройства и

предназначена для его сброса. На рис. 3.2 эта кнопка находится в правом верхнем углу, между разъемом микро-USB и разъемом питания.

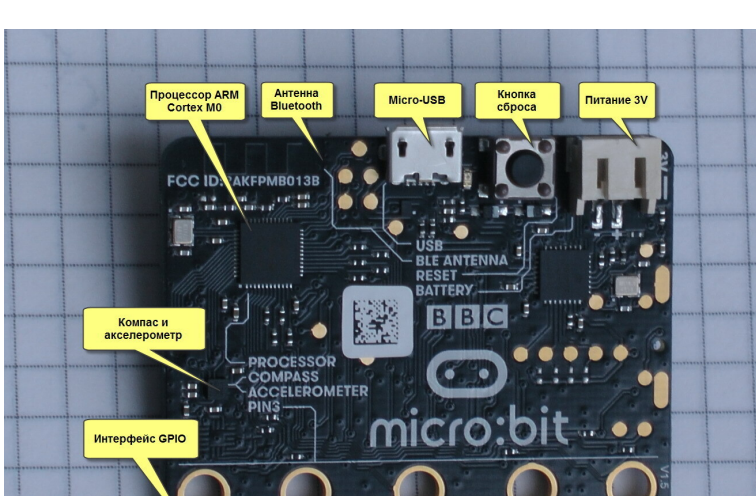


Рис. 3.2. Кнопка сброса находится на обратной стороне платы micro:bit

Две другие кнопки находятся на лицевой стороне платы микроконтроллера (рис. 3.3). Именно эти две кнопки, обозначенные как A и B, представляют для нас основной интерес.

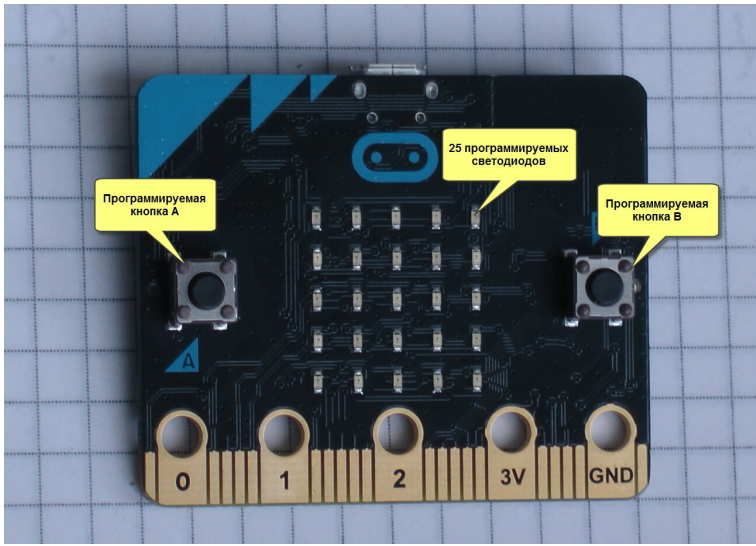


Рис. 3.3. Программируемые кнопки A и B на лицевой стороне платы micro:bit

Блоки программы, загруженной в память микроконтроллера, могут получать управление, когда нажимается одна из этих кнопок, а также когда они нажимаются вместе.

В микроконтроллере micro:bit версии 2 добавлена еще одна кнопка – сенсорная. Ее роль теперь играет логотип, который раньше представлял собой простой рисунок, расположенный над матрицей светодиодов (рис. 3.3). Теперь этот логотип превратился в сенсорную кнопку.

В конце этой главы будет приведена программа, работающая с сенсорным логотипом.

Если вы создаете дистанционно управляемую модель марсохода (ровера), то сможете использовать кнопки, например, для управления движением. Используйте одну из этих кнопок, например, для включения движения вперед, другую – для включения заднего хода. Одновременное нажатие кнопок пригодится, например, для экстренной остановки моторов ровера.

Задаем действия при нажатии кнопок

Для того чтобы проверить действие кнопок на практике, добавьте в новый проект из палитры **Ввод** три блока **кнопка нажата** (рис. 3.4).

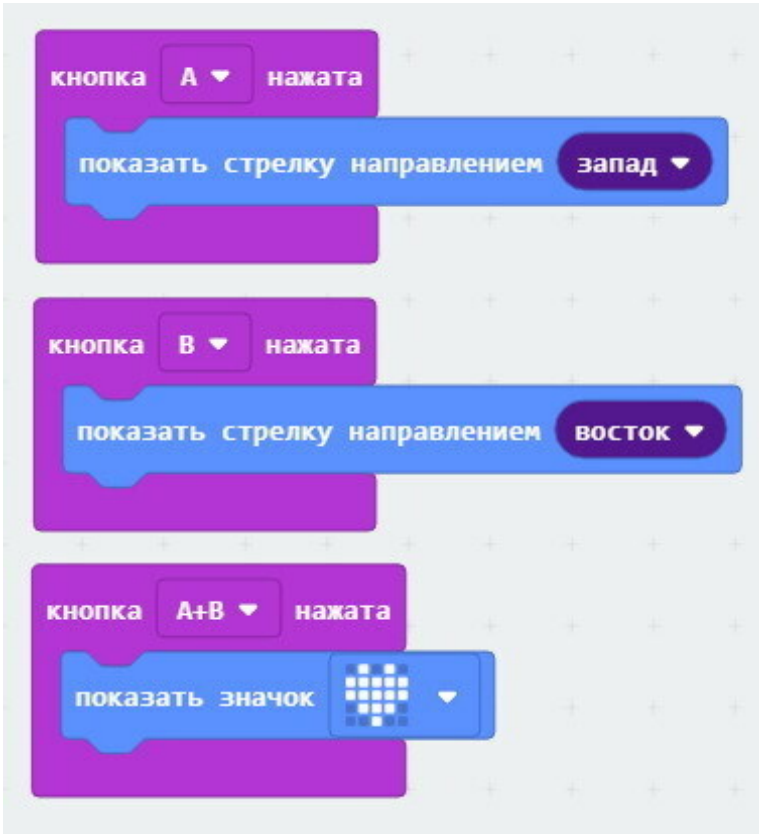


Рис. 3.4. Блоки для обработки событий от кнопок

Текст программы можно загрузить из архива, он записан в файл BoxRover/ch03/microbit-кнопки-влево-и-вправо.hex.

Когда вы нажимаете кнопки A и B, в микроконтроллере

возникают так называемые события. Эти события перехватываются блоками **кнопка нажата**. Если в тело этих блоков добавить другие блоки, то они получат управление при нажатии соответствующей кнопки, или двух кнопок одновременно.

Наша программа выполняет очень простые действия. Если нажать кнопку A, то программа нарисует на экране микроконтроллера стрелку, направленную на запад. Нажатие кнопки B приведет к тому, что будет нарисована стрелка, направленная на восток. И, наконец, если нажать обе кнопки одновременно, на экране будет нарисован значок сердечка (рис. 3.5).

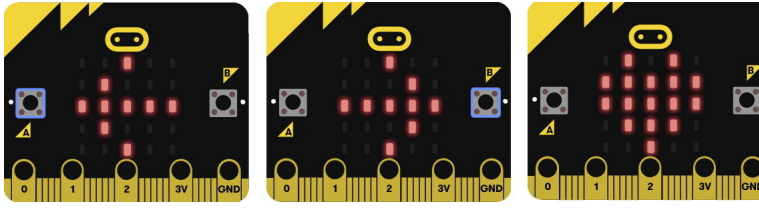


Рис. 3.5. Значки появляются при нажатии соответствующих кнопок

Обратите внимание, что мы не используем блоки **постоянно** и **при начале**. Обработка событий от нажатых кнопок выполняется блоками **кнопка нажата** независимо от работы блоков **постоянно** и **при начале**.

Для перехвата события создается так называемый программный хандлер (handler), или обработчик. Для одного события может быть создан только один обработчик.

Таким образом, вы можете создать только один обработчик для кнопки A, один – для кнопки B, и один для события, которое возникает при одновременном нажатии обеих кнопок.

Устройство, создающее событие, мы будем называть генератором событий. Как вы увидите дальше, помимо кнопок в микроконтроллере micro:bit существуют и другие генераторы событий, например, связанные с передачей данных по радио или через последовательный интерфейс UART.

Счетчик нажатий кнопок

Нашей следующей программой, работающей с кнопками A и B, будет несложный счетчик. Сразу после запуска значение счетчика будет равно нулю. При нажатии кнопки A значение счетчика увеличивается, а при нажатии кнопки B – уменьшается.

Программа счетчика показана на рис. 3.6 (файл BoxRover/ch03/microbit-Счетчик.hex).

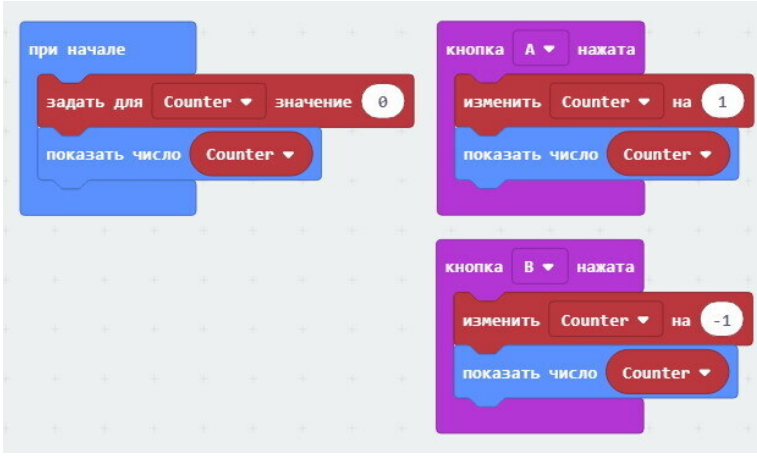


Рис. 3.6. Программа счетчика нажатий кнопок

Вначале из палитры **Основное** мы добавили блок **при**

начале. Кроме этого, в палитре **Переменные** мы создали переменную с именем **Counter**, предназначенную для хранения текущего значения счетчика.

В блоке **при начале** задается исходное значение счетчика **Counter**, равное нулю. После этого текущее значение счетчика показывается на экране в блоке **показать число**.

Когда нажимается кнопка A, блок **изменить** увеличивает значение **Counter** на единицу. Вслед за этим новое значение счетчика выводится на экран. Аналогично, при нажатии кнопки B значение счетчика нажатий уменьшается, и на экран выводится уменьшенное значение (рис. 3.7).

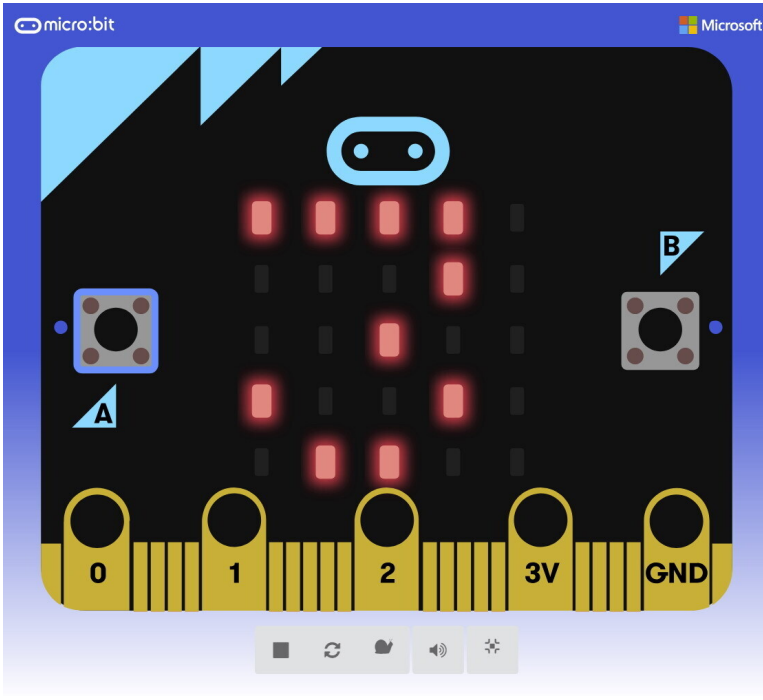


Рис. 3.7. Текущее значение счетчика нажатий отображается на мониторе

Обратите внимание, что если числовое значение превысит 9, числа будут показываться на экране micro:bit в режиме бегущей строки. Режим бегущей строки будет включен и для отображения отрицательных чисел.

Проверка состояния кнопки

Предыдущие программы выполняли какое-либо действие, когда мы нажимали кнопки. Но есть и другая возможность – в процессе своей работы программа может проверять текущее состояние кнопок, и в зависимости от результатов проверки изменять свое поведение.

В качестве примера доработаем программу, взятую из раздела **Доверимся случаю** предыдущей главы, которая зажигает и гасит светодиоды с различной яркостью. Сделаем так, чтобы с помощью кнопки A можно было запускать программу и ставить ее на паузу. А именно, звезды будут зажигаться и гаснуть только если кнопка A нажата. В противном случае вся жизнь в нашей вселенной останавливается.

Код программы показан на рис. 3.8. Вы найдете эту программу в файле microbit-Вселенная-кнопка-A.hex.

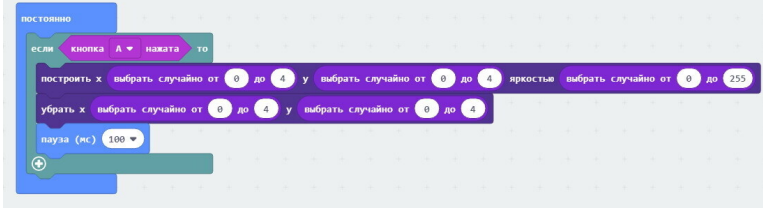


Рис. 3.8. Программа работает только при нажатой

кнопке A

Здесь мы добавили в **постоянно** блок **если**. В качестве условия в блок **если** мы вставили проверку **кнопка нажата**. Эта проверка находится в палитре **Ввод**.

Давайте усложним нашу программу. Пусть теперь звезды зажигаются и гаснут, если кнопка A не нажата, и перестают зажигаться и гаснуть, когда мы нажимаем кнопка A.

Для кнопки B тоже найдется работа. Каждый раз когда мы будем ее нажимать, процесс зажигания и угасания звезд должен замедляться.

Новый вариант программы показан на рис. 3.9, файл microbit-Вселенная-на-паузе.hex.

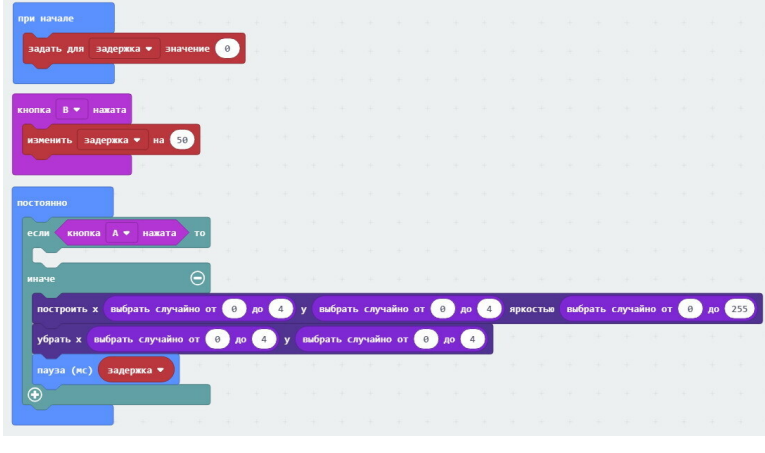


Рис. 3.9. Новая версия программы управления звез-

дами

В блоке **при начале** мы задаем в переменной **задержка** начальное значение задержки, равное 0. При нажатии на кнопку B мы увеличиваем значение задержки на 50 мс.

В блоке **постоянно** мы использовали цикл **если иначе**. Как он работает?

В теле условия **если** ничего нет, поэтому если кнопка A нажата, то никакие блоки не выполняются и бесконечный цикл работает вхолостую. Но если кнопка A не нажата, то в дело включается тело **иначе**. Здесь у нас находится блоки, управляющие переключением звезд, а также блок паузы.

Как мы уже говорили, по умолчанию значение параметра для блока **пауза**, которое хранится в переменой **задержка**, равно нулю. Если вы будете нажимать кнопку B, то это значение будет увеличиваться на 50 мс при каждом нажатии. В результате жизнь наших звезд будет каждый раз замедляться.

Сенсорная кнопка в виде логотипа

В micro:bit версии 2 появилась сенсорная кнопка в виде логотипа (рис. 3.10). Ее можно нажимать, дотрагиваясь пальцем.

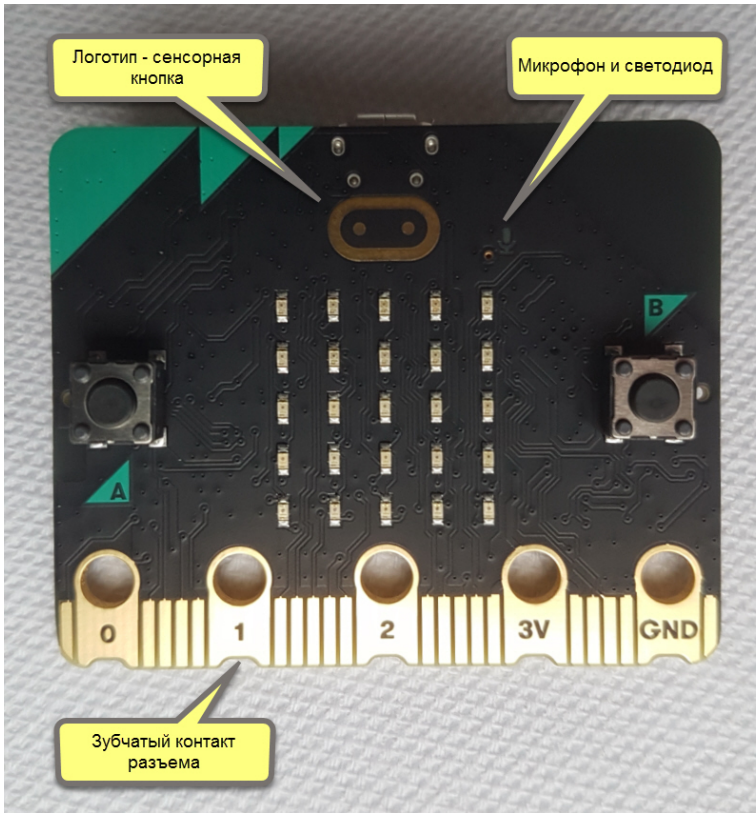


Рис. 3.10. Логотип стал сенсорной кнопкой

Для работы с сенсорной кнопкой в виде логотипа в палитре **Ввод**, в разделе **micro:bit (V2)** появились блоки **on logo** и **logo is pressed** (рис. 3.11).

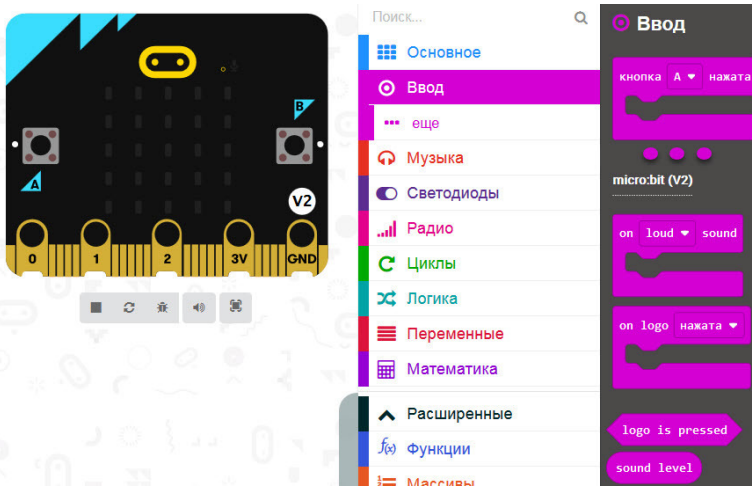


Рис. 3.11. Блоки для работы с сенсорной кнопкой в виде логотипа

Блок **on logo** получает управление, если дотронуться до логотипа пальцем. С помощью блока **logo is pressed** можно проверить, нажат логотип в настоящее время, или нет.

На рис. 3.12 мы показали простейшую программу microbit2-Сенсорный-логотип.hex, реагирующую на нажатие логотипа.

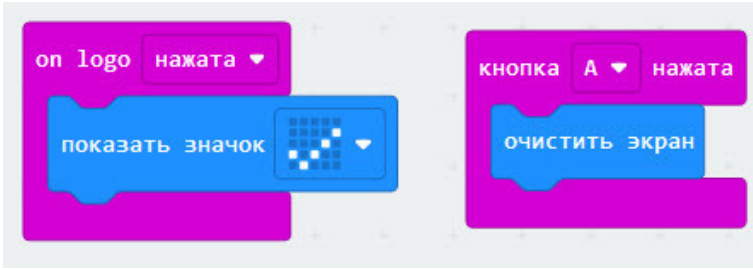


Рис. 3.12. Обработка событий от логотипа

Если нажать логотип, на экране micro:bit появится галочка. Чтобы она исчезла, нажмите кнопку **A**.

В блоке **on logo** можно указать, что событие от логотипа должно возникать только в том случае, если нажимать на него достаточно долго. Такое поведение демонстрирует программа microbit2-Логотип-нажат-долго.hex (рис. 3.13).

Рис. 3.13. Реагирование на длительное нажатие логотипа

Вы также можете регистрировать события, возникающие в момент отпускания логотипа. Для этого в меню блока **on logo** выберите строку **отжата**.

Еще одна несложная программа microbit2-Сенсорный-логотип-2.hex демонстрирует использование блока **logo is pressed** (рис. 3.14).

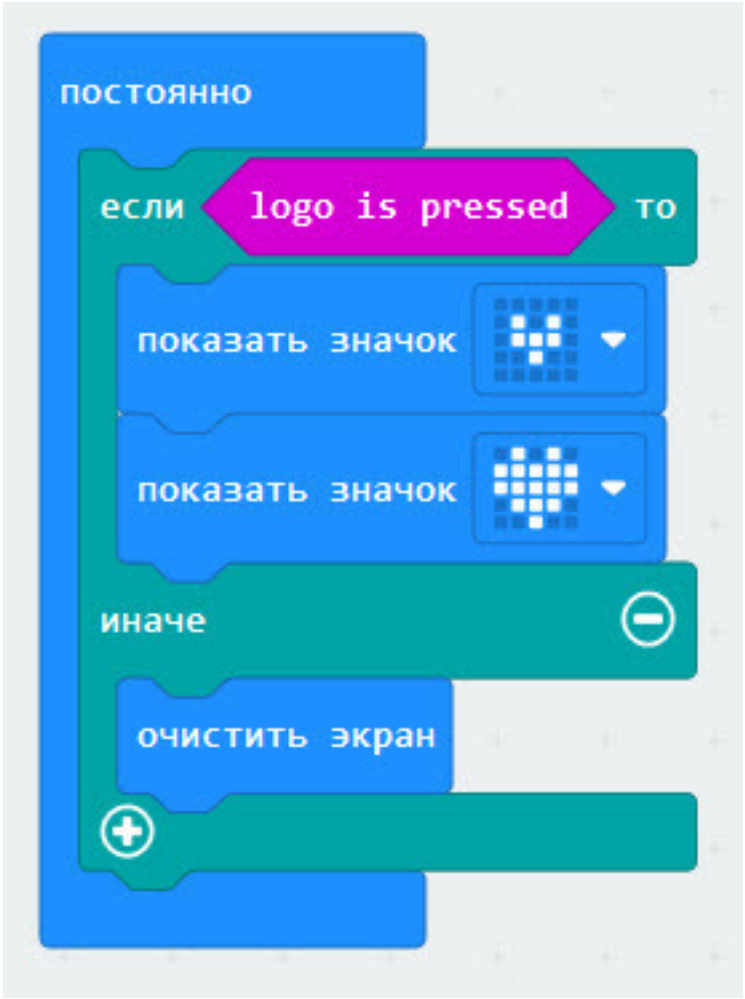

постоянно
если
logo is pressed
то
показать значок
показать значок
иначе
очистить экран

Рис. 3.14. Проверяем нажатие логотипа в цикле

Эта программа проверяет состояние кнопки логотипа в цикле. Пока логотип нажат, программа рисует на экране micro:bit бьющееся сердце. Но как только вы отпустите логотип, сердце исчезает!

Сенсорные контакты

Как мы уже говорили, во многих приборах используются сенсорные кнопки. Такие кнопки служат долго, т.к. в них нет механических движущихся деталей.

Сенсорные кнопки на micro:bit версии 2 могут работать в емкостном и резистивном режиме.

В емкостном режиме используется тот факт, что прикосновение к кнопке равносильно подключению к ней конденсатора.

В резистивном режиме предполагается, что человек касается одновременно кнопки и общего провода (земли, минуса источника питания). При этом изменяется сопротивление между кнопкой и землей от практически бесконечного до значений, исчисляемых единицами или десятками КОм.

Забегая вперед, заметим, что вы можете использовать в качестве сенсорных контактов выводы P0, P1 и P2 в качестве сенсорных кнопок. На рис. 3.11 эти контакты отмечены цифрами 0, 1 и 2. Подробнее мы расскажем о контактах разъема платы micro:bit в 7 и 8 главе нашей книги.

На рис. 3.15 мы показали программу microbit2-Сенсорный-контакт.hex, которая делает то же самое, что и программа microbit2-Сенсорный-логотип-2.hex.

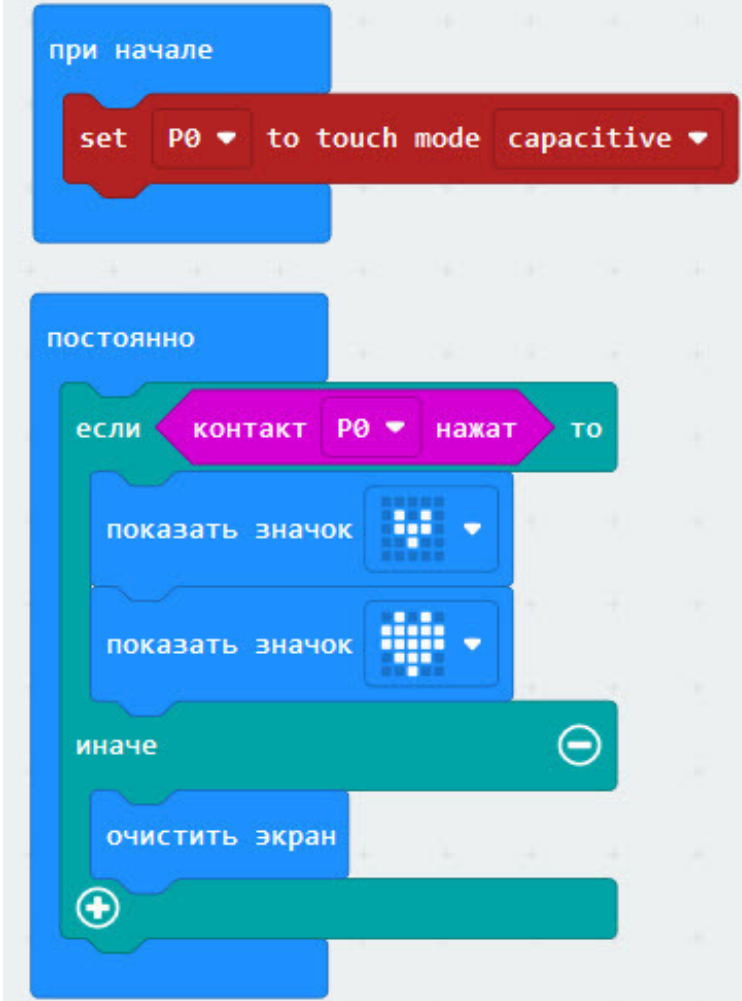

при начале
set P0 to touch mode capacitive
постоянно
если контакт P0 нажат то
показать значок
показать значок
иначе
очистить экран

Рис. 3.15. Используем контакт P0 в качестве сенсорной кнопки

В начале своей работы программа устанавливает один из двух возможных режимов работы сенсорного контакта – емкостной или резистивный. Чтобы использовать емкостной режим, в меню блока **set P0 to touch mode** выберите строку capacitive, а чтобы в резистивном – строку **resistive**.

Блок **set P0 to touch mode** находится в палитре **Расширенные**, **Контакты**, **еще**, в разделе **micro:bit (v2)**.

Загрузив программу в микроконтроллер, дотроньтесь пальцем до контакта, обозначенного на плате micro:bit цифрой 0. Пока вы держите палец, на экране micro:bit будет биться сердце.

Домашнее задание

В качестве **первого домашнего задания** попробуйте самостоятельно доработать программу microbit-Счетчик.hex (рис. 3.6) таким образом, чтобы при нажатии двух кнопок одновременно текущее значение счетчика сбрасывалось в нулевое значение.

Решение вы сможете найти в файле microbit-Счетчик-со-сбросом.hex, загрузив архив программ с сайта автора этой книги http://frolov-lib.ru/books/boxrover/.

В качестве **второго самостоятельного задания** добавьте к программе microbit-Вселенная-на-паузе.hex (рис. 3.9) код, который сбрасывает задержку к исходному значению при одновременном нажатии кнопок A и B.

Решение этого задания есть в файле microbit-Вселенная-на-паузе-со-сбросом.hex.

В качестве **третьего домашнего задания** попробуйте сделать кодовый замок. Замок должен работать следующим образом.

Чтобы открыть замок, вам нужно нажать четыре раза кнопку A, и один раз – кнопку B (в любой последовательности). Только эта комбинация должна открыть замок.

Для проверки состояния замка нажмите кнопки A и B одновременно. Если замок открылся, на экране micro:bit должно быть нарисовано сердечко, если нет, то крестик.

Решение третьего задания вы найдете в файле microbit-Секретный-счетчик.hex.

И, наконец, в **четвертом домашнем задании** попробуйте изменить режим **set P0 to touch mode** с емкостного на резистивный. Учтите, что это задание нужно выполнять на micro:bit версии 2.

Программа с измененным режимом записана в файле microbit2-Сенсорный-контакт-2.hex. Проверьте, есть ли разница в работе этой программы при изменении режима.

Итоги

Во третьей главе мы познакомились с обработкой событий от кнопок A и B, расположенных на лицевой стороне платы микрокомпьютера micro:bit, а также от сенсорной кнопки micro:bit версии 2, роль который играет логотип. Вы научились проверять текущее состояние кнопок во время работы программы.

Вы научились задавать действия при нажатии кнопок A и B, а также в тех случаях, когда эти кнопки были нажаты одновременно. Кроме того, вы теперь можете использовать контакты P0, P1 и P2 в качестве сенсорных кнопок.

Также вы усложнили программу зажигания звезд на экране микрокомпьютера и сделали так, что ее работа стала зависеть от текущего состояния кнопок.

При выполнении домашних заданий вы доработали программу управления счетчиком и звездами в вашей микро-вселенной, создали кодовый замок и испытали два разных режима работы сенсорных контактов – емкостной и резистивный.

4. Измеряем температуру

Мы уже писали во введении к этой книге, что «на борту» микроконтроллера micro:bit имеется измеритель температуры. Физически он находится внутри процессора, и поэтому, строго говоря, измеряет не температуру окружающего воздуха, а температуру самого процессора.

Так как процессор micro:bit потребляет очень малую энергию, то в первом приближении можно считать, что его температура примерно соответствует температуре окружающей среды. Конечно, для более точных измерений не обойтись без специального внешнего термометра (и мы его подключим позже, когда займемся моделью марсохода BoxRover), но сейчас для нас будет вполне достаточно измерителя температуры, интегрированного в micro:bit.

Как работает измеритель температуры

Здесь, конечно, нас не будут интересовать ртутные или иные аналоговые градусники. Нам интересно узнать, как измеряется температура в электронных устройствах.

Для измерения температуры применятся так называемые терморезисторы. Терморезистор представляет собой полупроводниковый прибор, сопротивление которого зависит от температуры.

Зависимость эта нелинейная, однако можно выставить рабочую точку терморезистора так, что она попадет на линейный участок. В этом случае по изменению сопротивления терморезистора можно будет судить об изменении его температуры.

В простейшем случае можно подключить терморезистор к батарейке через миллиамперметр, и проследить за изменением проходящего тока при нагреве или охлаждении терморезистора (рис. 4.1).

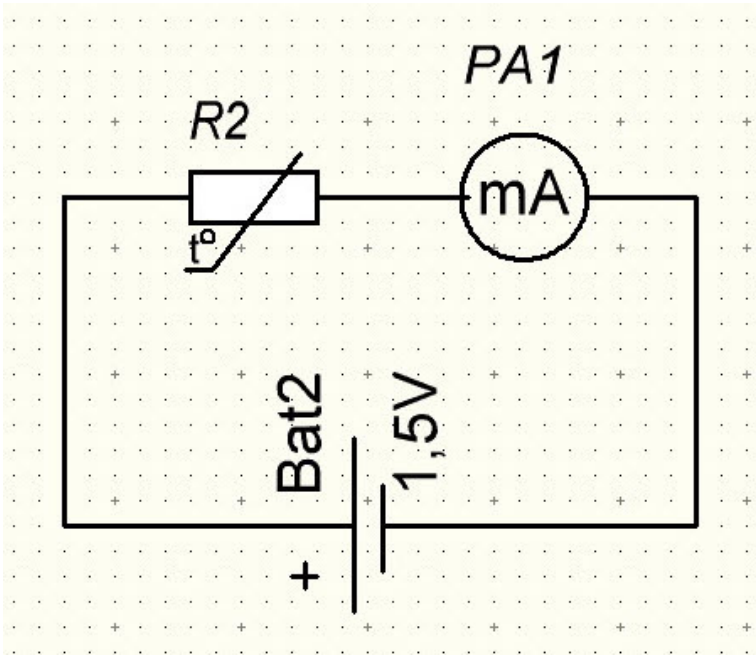


Рис. 4.1. Включение терморезистора

Конечно, настоящие схемы измерения температуры совсем не такие, но для понимания принципа и этого будет достаточно.

Вы также можете контролировать изменение сопротивления терморезистора с помощью омметра или тестера, переключенного в режим измерения сопротивления.

Существуют два типа терморезисторов. Это термисторы

и позисторы. Сопротивление термистора уменьшается при увеличении температуры, а позистора наоборот, увеличивается. Говорят, что термисторы обладают отрицательным температурным коэффициентом сопротивления (ТКС), а позисторы – положительным.

В интернете есть множество статей, посвященных терморезисторам, например, эта – https://elektrikexpert.ru/termorezistor.html. На данном этапе мы не будем углубляться в детали, т.к. для измерения температуры будем использовать готовые устройства, учитывающие все особенности, в частности, нелинейную зависимость сопротивления терморезистора от температуры.

Термометр из micro:bit

Вы можете очень просто превратить свой микроконтроллер micro:bit в термометр. Достаточно использовать программу, показанную на рис. 4.2.

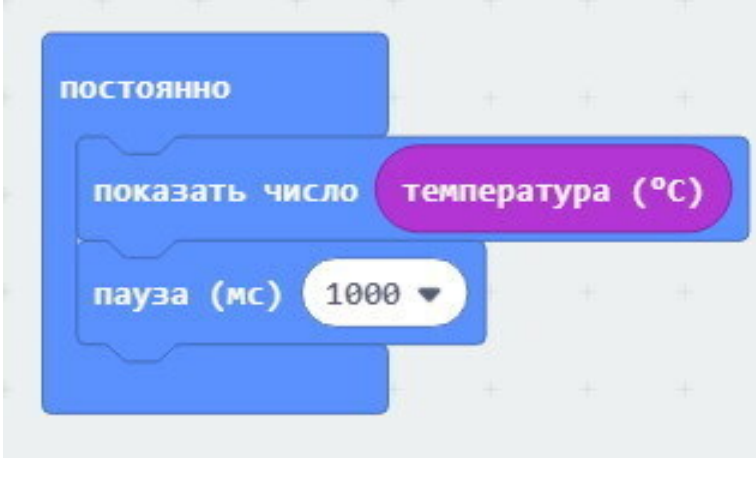


Рис. 4.2. Программа для измерения температуры процессора micro:bit

Эта программа находится в файле BoxRover/ch04/microbit-измеритель-температуры.hex.

Здесь мы добавили в блок **постоянно** блок **показать число**, предназначенный для вывода числа на экран микро-

компьютера, а также блок задержки на одну секунду. В качестве значения мы вставили из панели **Ввод** блок **температура** (**°C**). Этот блок возвращает значение температуры процессора micro:bit в градусах Цельсия.

Таким образом, на экран раз в секунду выводится текущее значение температуры в виде бегущей строки (рис. 4.3).

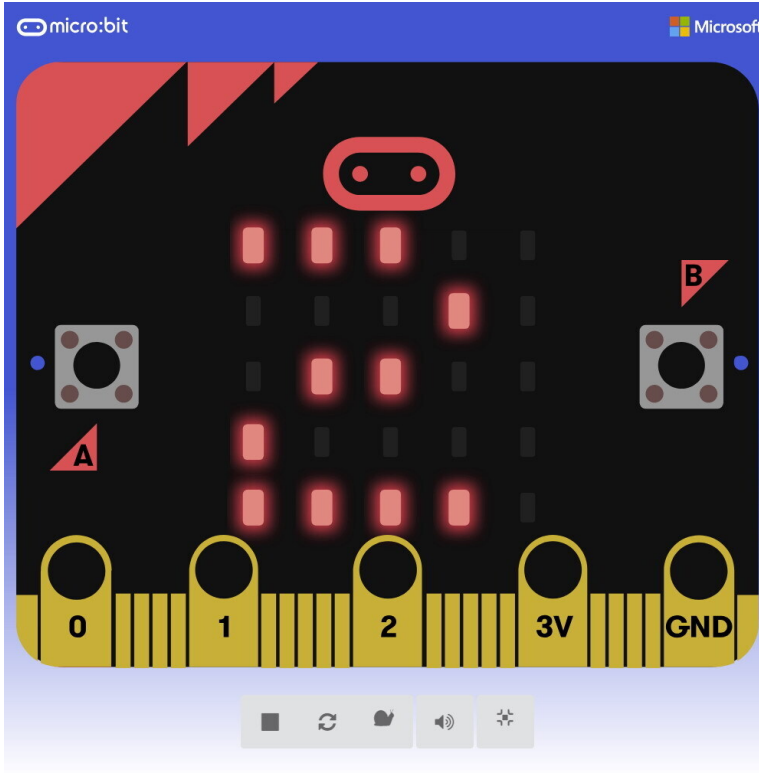


Рис. 4.3. На монитор micro:bit выводится текущая температура процессора

Как можно использовать встроенный измеритель температуры?

Например, с его помощью можно контролировать перегрев процессора микрокомпьютера. Если температура превысила максимально допустимое значение, можно предпринимать какие-либо действия.

Подготовим программу, которая при превышении температуры сверх заданного значение (30 °C) включает сразу все светодиоды экрана micro:bit. Если же температура нормальная, на экране будет нарисовано сердце (рис. 4.4).

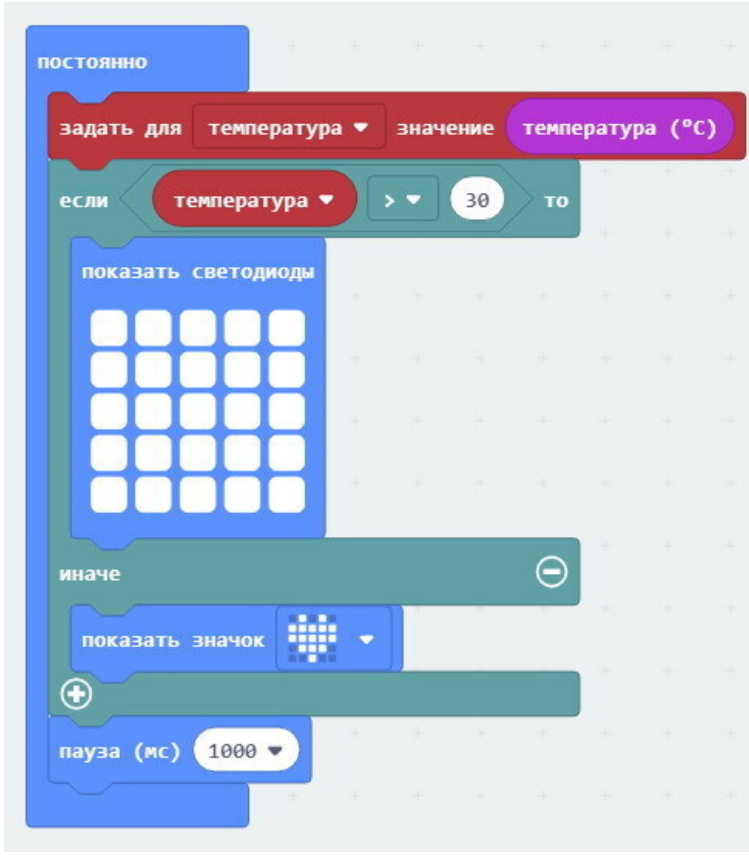


Рис. 4.4. Программа контроля температуры

Программа сохранена в файле microbit-проверка-температуры.hex.

Здесь в палитре **Переменные** мы определили перемен-

ную с именем **температура**. При помощи блока **задать** мы задаем значение этой переменной, равной блоку **температура (°C)**, добавленному из панели **Ввод**.

Когда срабатывает условие, что значение, записанное в переменную температура, превышает 30 °C, блок **показать светодиоды** включает на экране все светодиоды сразу.

Если же перегрева нет, то блок **показать значок** выводит на экран изображение сердца.

Домашнее задание

Попробуйте поместить плату micro:bit на пару минут в холодильник (только не в морозильную камеру), а также оставить на солнце. Посмотрите, как это скажется на результатах измерения температуры процессора.

Проследите, чтобы в ходе экспериментов с холодильником на плату micro:bit не попала влага.

Учтите, что значительны перегрев платы micro:bit, а также переохлаждение могут привести к выходу платы из строя. Согласно описанию https://tech.microbit.org/hardware/, датчик температуры в micro:bit работает в диапазоне от -25C до 75C, а его точность составляет ±4C.

Итоги

В этой главе вы узнали, что в микроконтроллере micro:bit имеется встроенный измеритель температуры. И хотя он установлен непосредственно внутри микропроцессора и показывает его температуру, а не температуру окружающей среды, все равно он может принести определенную пользу. Например, вы можете обнаружить перегрев процессора, опасный для работы вашей модели марсохода BoxRover или модуля автоматизации умного дома.

Вы составили программу, которая показывает текущее значение температуры, а также программу контроля превышения заданного значения температуры.

В следующих главах книги вы научитесь подключать к micro:bit миниатюрную погодную станцию, способную показывать не только температуру окружающей среды, но и давление, влажность, а также температуру точки росы.

5. Измеряем ускорение и контролируем перегрузки

Микроконтроллер micro:bit оборудован очень интересным устройством – акселерометром, или измерителем ускорения.

Вспомним знания, полученные еще в школе. Ускорением называется быстрота изменения скорости тела. Например, когда вы сбрасываете кирпич с крыши дома (никогда так не делайте), то его скорость будет увеличиваться с ускорением свободного падения, равном примерно 9,8 м/с^2. Это означает, что каждую секунду скорость падающего кирпича будет увеличиваться на 9,8 м/с.

Обычно при вычислениях ускорение свободного падения обозначается как g.

Обратите внимание, что ускорение свободного падения не зависит от массы тела. Иначе говоря, не важно, сбрасываете ли вы тяжелый кирпич, или легкий металлический шарик, ускорение будет одинаковым. Другое дело, что на изменение скорости может повлиять сопротивление воздуха (надувной шарик может вообще полететь не вниз, а вверх), но все это вам известно из школьного курса физики.

Когда вы летите в самолете, то можете испытывать ускорение (или иными словами, перегрузки) до 1,5g. Парашю-

тисты и космонавты подвержены более серьезным перегрузкам, порядка 4g и более. Очень большие перегрузки возникают, когда вы едете в автомобиле и он на ходу врезается в стену или в автомобиль, который едет вам навстречу. Значительные перегрузки, как ударные, так и постоянно действующие, могут привести к серьезным проблемам в здоровье.

Принцип измерения ускорения

Наверное вы уже поняли, как важно контролировать ускорение, например, при испытании различной техники. Но как измеряется ускорение?

В микрокомпьютере micro:bit используются так называемый емкостной акселерометр. Его принцип действия иллюстрируется на рис. 5.1.

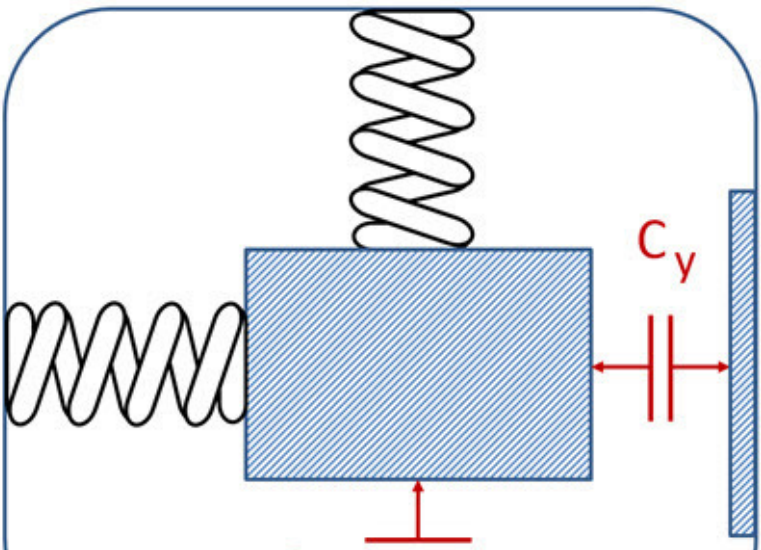


Рис. 5.1. Иллюстрация принципа действия емкостного акселерометра

Представьте себе, что в пластмассовой коробке на двух пружинах закреплен небольшой металлический груз. Снизу и справа от груза имеются две металлические пластины.

Когда корпус перемещается с ускорением вверх или вниз, вправо или влево, расстояние между грузом и пластинами будет изменяться. Это происходит из-за инертной массы гру-

за, закрепленного на гибкой пластине.

Груз и пластины образуют два конденсатора, обозначенные на рис. 5.1 как C_X и C_Y. Когда расстояние между грузом и пластинами изменяется, меняются и емкости этих конденсаторов.

Таким образом, измеряя емкости конденсаторов в процессе движения, можно отслеживать изменение ускорения по горизонтальной и вертикальной оси. А если добавить третью пружину и еще одну металлическую пластину, то можно будет измерять ускорение во всех трех направлениях.

Что касается акселерометра, встроенного в micro:bit, то там применяется три пластины, что дает возможность измерять ускорение по всем трем осям координат.

Согласно документации, акселерометр micro:bit может измерять ускорение в диапазоне от -2g до 2g. При этом возвращаются положительные и отрицательные значения в тысячных долях g, с диапазоном от 0 до 1024.

Обнаружение жестов

Одно из самых интересных применений акселерометра micro:bit – это обнаружение жестов. Это поворачивание платы микроконтроллера логотипом вверх или вниз, наклон вправо или влево, поворачивание монитором вверх или вниз, встряхивание, свободное падение, а также движение с ускорением в 3g, 6g и 8g.

Когда вы совершаете с платой микроконтроллера описанные выше действия, создаются соответствующие события. Их можно обрабатывать в блоке **по жесту** из палитры **Ввод**. Список доступных жестов показан на рис. 5.2.

Рис. 5.2. Жесты в микрокомпьютере micro:bit

Расскажем, как вы можете обрабатывать жесты от акселерометра в своих программах.

Бросаем кости

Самое простое что можно придумать с жестами – это сделать игру в кости. Когда вы встряхиваете плату своего micro:bit, на его экране высвечивается случайное число от 0 до 6. Конечно, это упрощенный вариант игры, так как тут мы «бросаем» только одну кость.

Программа игры в кости показана на рис. 5.3.

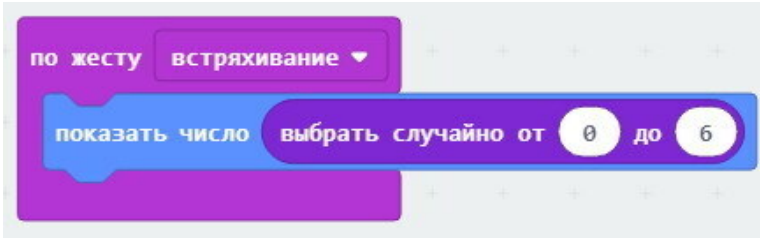


Рис. 5.3. Программа обработки события встряхивания

Когда микроконтроллер обнаруживает, что его начали трясти, он создает событие на жест **встряхивание**. Обработчик этого события показывает на экране случайное число в диапазоне от 0 до 6.

Эта программа находится в файле BoxRover/ch05/microbit-Кости.hex архива, который можно скачать с сайта

автора книги по адресу http://frolov-lib.ru/books/boxrover/.

Отслеживаем ориентацию платы micro:bit в пространстве

Когда вы работаете со смартфоном, то независимо от его положения в пространстве просматриваемая страница поворачивается таким образом, что вы никогда не видите ее вверх ногами. Акселерометр, встроенный в смартфон, позволяет все время держать на контроле положение экрана смартфона, отслеживая повороты экрана.

Давайте создадим такую программу для micro:bit, которая будет рисовать стрелку, показывающую вверх независимо от того, как вы держите плату микроконтроллера.

Эта программа показана на рис. 5.4.

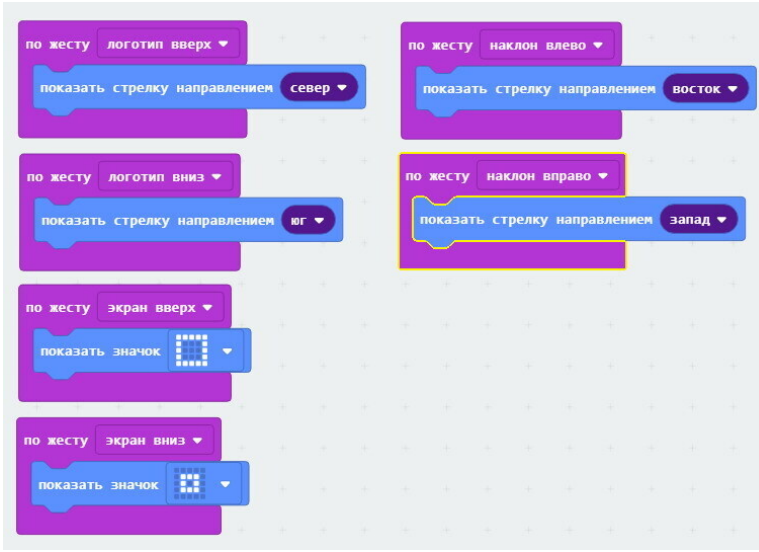


Рис. 5.4. Программа для отслеживания положения платы micro:bit в пространстве

Как видите, программа обрабатывает шесть жестов. Это поворот платы микроконтроллера логотипом вверх, вниз, наклон вправо или влево, а также расположение платы экраном вверх и экраном вниз.

Каждый раз, когда возникает событие, связанное с тем или иным жестом, программа при помощи блока **показать стрелку направлением** рисует стрелку таким образом, что она всегда показывает вверх.

Если вы кладете контроллер горизонтально на стол экра-

ном вверх, на экране будет нарисован квадрат большого размера. Если же вы перевернете плату контроллера экраном вниз, то увидите на экране квадрат маленького размера.

Все возможные варианты показаны на рис. 5.5.

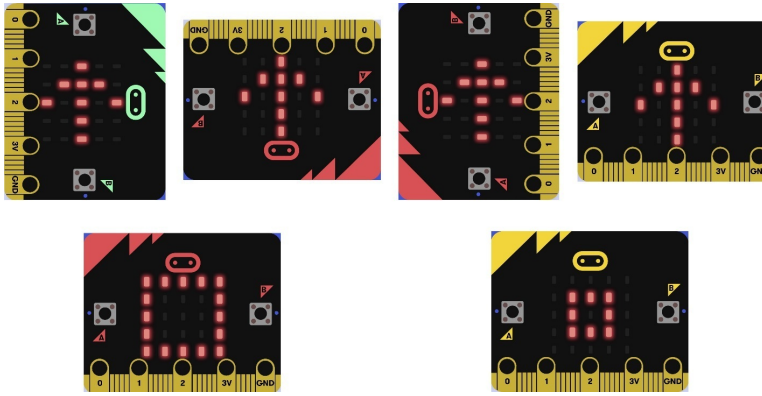


Рис. 5.5. Изображение на экране контроллера зависит от его расположения в пространстве

Программа находится в файле BoxRover/ch05/microbit-Жесты.hex.

Обнаружение невесомости и перегрузок

Если ваш микроконтроллер micro:bit находится в космосе, или просто упал со стола, то возникнет событие от жеста **свободное падение**.

На рис. 5.6 показана очень простая программа, которая может обнаружить состояние невесомости.

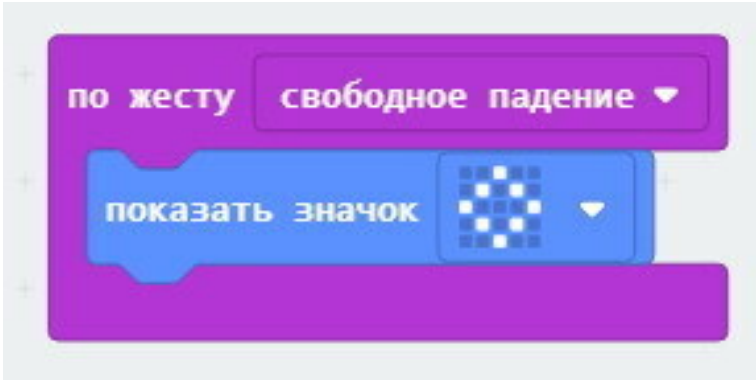


Рис. 5.6. Программа обнаружения невесомости

Когда ваш micro:bit окажется в невесомости, на его экране будет нарисован ромб (рис. 5.7).

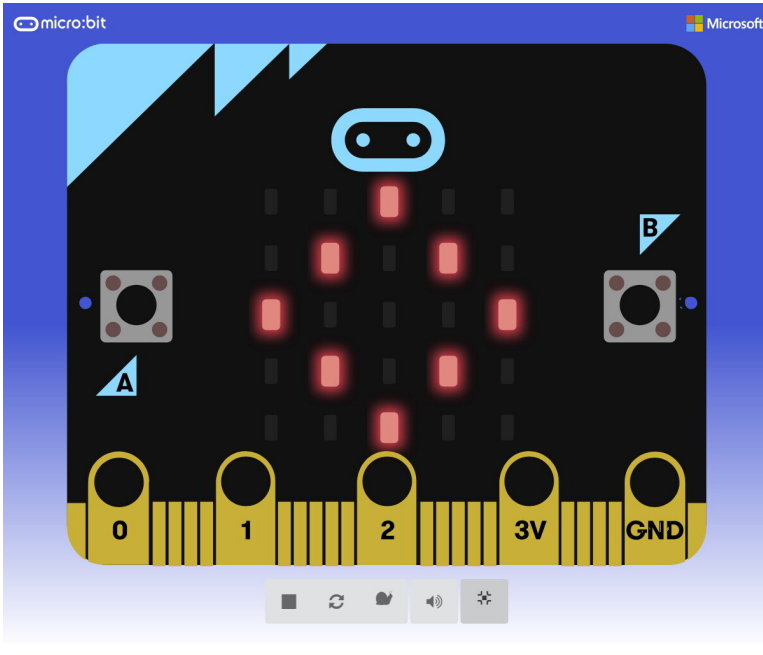


Рис. 5.7. Обнаружено состояние невесомости

Код программы мы сохранили в файле BoxRover/ch05/microbit-Невесомость.hex.

Испытывая программу, соблюдайте осторожность чтобы не повредить ваш micro:bit. Вы можете бросать его на что-то мягкое, на подушку или диван, либо просто ловить рукой.

Невесомость не всегда означает опасность, но вот перегрузки – совсем другое дело. Давайте создадим программу, которая обнаруживает перегрузки из-за слишком силь-

ного ускорения (рис. 5.8). Эта программа записана в файл BoxRover/ch05/microbit-Перегрузка.hex.

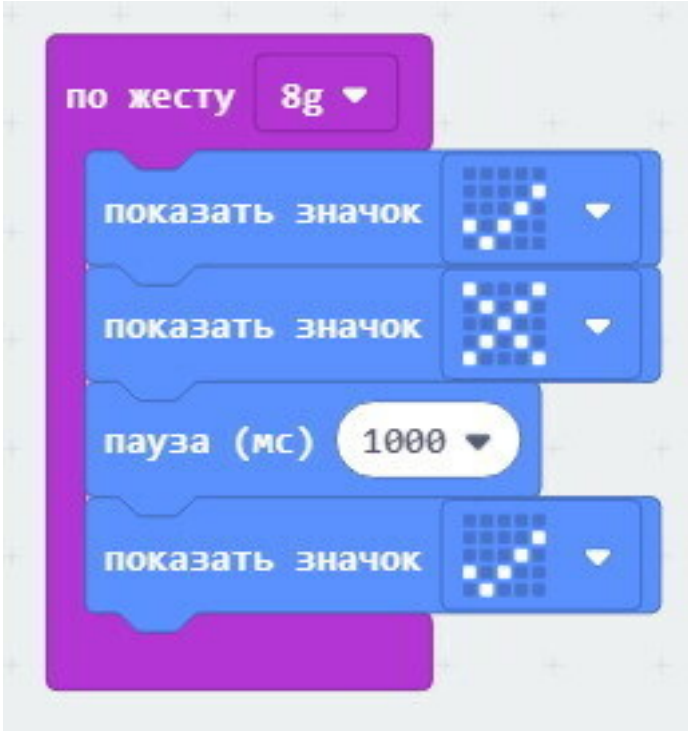


Рис. 5.8. Программа обнаружения перегрузок

Когда будет обнаружена перегрузка свыше 8g, на экране сначала будет нарисована галочка, потом на одну секунду — крестик (рис. 5.9), и затем снова галочка.

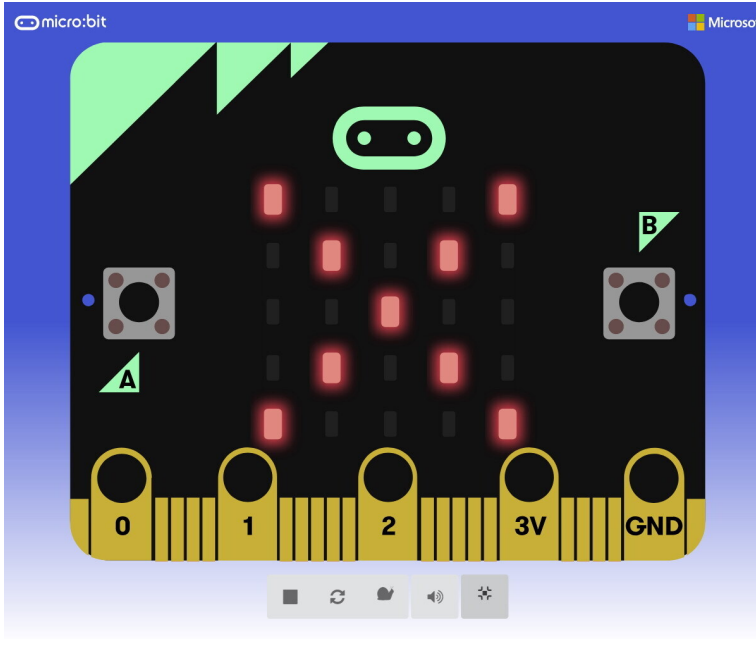


Рис. 5.9. Обнаружена перегрузка

Вы можете менять значение порога ускорения в блоке по жесту, указывая там 3g, 6g или 8g.

Но **будьте осторожны**, подвергая плату микроконтроллера перегрузкам – она может выйти из строя. Для проверки работы программы возьмите плату micro:bit в руку и встряхните. Если плата подключена к компьютеру через разъем микро-USB, не держите ее за кабель, чтобы не повре-

дить разъем. Прикладывайте усилия таким образом, чтобы не сломать разъем на плате контроллера.

Рисуем гистограмму значений ускорения

Возможно, вы никогда не сталкивались с понятием гистограммы. Расскажем кратко, что это такое.

Гистограммой называется графический способ представления табличных данных. При построении графика гистограммы по горизонтальной оси откладываются интервалы, в которые могут попадать значения. В нашем случае мы будем использовать здесь значения ускорения, полученные от акселерометра micro:bit.

По вертикальной оси откладывается частота попадания значений в тот или иной интервал. При этом площадь прямоугольника, соответствующая интервалам горизонтальной оси, соответствует количеству значений, попавших в данный интервал.

Перейдем от теории к практике.

Подготовим программу, которая строит гистограмму значений, полученных от акселерометра микроконтроллера, на его экране (рис. 5.10). Она находится в файле BoxRover/ch05/microbit-Гистограмма-ускорения.hex.

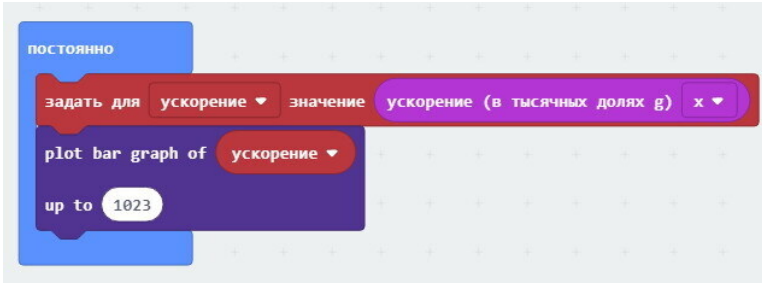


Рис. 5.10. Построение гистограммы значений ускорения

Здесь мы определяем переменную **ускорение**, после чего запускается бесконечный цикл. В цикле в эту переменную записывается значение ускорения в тысячных долях g, полученное от акселерометра micro:bit.

Далее управление получает блок **plot bar graf**, добавленный из палитры **Светодиоды**.

В качестве параметра мы передаем этому блоку текущее значение ускорения, полученное от акселерометра. В параметра **up to** нужно указать максимальное значение, которое может вернуть акселерометр. Мы знаем, что это значение равно 1023.

Теперь загрузим программу в память микроконтроллера и оставим его плату спокойно лежать на столе. На экране будет картина, показанная слева на рис. 5.11.

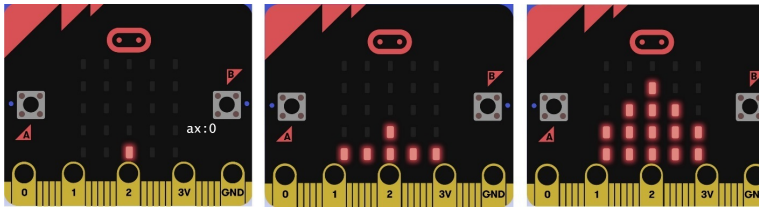


Рис. 5.11. Гистограммы значений от акселерометра

Если потихоньку трясти плату микрокомпьютера, то вы увидите быстро меняющуюся гистограмму (средняя и правая картинка на рис. 5.11). Чем сильнее вы будете трясти плату micro:bit, тем больше будет высота линий, нарисованных на экране.

Домашнее задание

В качестве **первого домашнего задания** доработайте программу бросания костей BoxRover/ch05/microbit-Кости.hex (рис. 5.3). Сделайте так, чтобы через секунду после встряхивания на экране micro:bit по очереди показывалось два случайных числа (как будто мы бросаем две кости). При этом сначала должно показываться одно случайное число, потом символ ромбика, а затем – второе случайное число.

Решение вы найдете в файле BoxRover/ch05/microbit-Бросаем-кости.hex.

Вторым домашним заданием будет сделать прототип пульта управления вашим марсоходом. Для управления движением ровера используйте кнопки A и B, а также жесты.

Результатом выполнения любой команды должно быть отображение кода команды на экране micro:bit. Вот список кодов команд:

0 – движение вперед при нажатии кнопки A или жесте «логотип вниз»;

1 – движение назад при нажатии кнопки B или жесте «логотип вверх»;

2 – поворот вправо при жесте «наклон вправо»;

3 – поворот влево при жесте «наклон влево»;

4 – останов при одновременном нажатии кнопок A и B или жесте «экран вверх»;

Пока вы получите только коды команд, но в дальнейшем эти коды будут отправляться по радио с пульта управления в ваш ровер, и будут использованы для управления его моторами.

Решение второго домашнего задания находится в файле BoxRover/ch05/microbit-Прототип-пульта.hex.

Итоги

В этой главе мы познакомились с очень интересным устройством, расположенным на плате микроконтроллера micro:bit – акселерометром.

С помощью акселерометра вы научились обнаруживать и обрабатывать события, связанные с жестами – поворот платы микроконтроллера логотипом вверх или вниз, наклон вправо или влево, поворот монитором вверх или вниз, встряхивание, свободное падение, а также движение с ускорением в 3g, 6g и 8g.

Вы запустили программу, которая показывает стрелку, указывающую всегда вверх, независимо от положения платы микрокомпьютера. Кроме этого, вы познакомились со способом обнаружения невесомости и перегрузок, а также научились строить гистограмму значений ускорения.

В качестве домашнего задания вы сделали программу для игры в кости, а также прототип пульта управления моделью марсохода. Ваш пульт управления использует не только кнопки, расположенные на плате micro:bit, но и жесты.

6. Встроенный магнитометр

В детстве вы наверняка играли с магнитами и знаете, что они могут притягиваться и отталкиваться друг от друга. Магниты притягиваются к железу, но игнорируют алюминий, дерево и другие материалы, которые называются немагнитными.

У магнита есть северный и южный полюса. Если взять два магнита, то они будут притягиваться друг к другу разными полюсами и отталкиваться, если вы попытаетесь прижать их друг другу одинаковыми полюсами.

Наша планета Земля тоже представляет собой огромный магнит! И у нее тоже есть северный и южный магнитные полюса, которые, кстати, не совсем совпадают с географическими полюсами.

Для того чтобы ориентироваться на местности, до появления спутниковых систем позиционирования, таких как GPS и ГЛОНАСС, путешественникам приходилось пользоваться магнитным компасом и бумажной картой.

Стрелка компаса представляет собой магнит в виде стрелки. Один конец стрелки покрашен синей краской и всегда показывает на сервер, а другой – красной краской и смотрит на юг. С помощью компаса можно сориентировать карту таким образом, чтобы ее верхний край был направлен в сторону сервера. После этого, привязавшись к местности, можно

будет понять, в какую сторону следует идти, чтобы не заблудиться.

Отправляя ровер на другую планету, вам будет полезно уметь ориентироваться на местности. Кроме того, измеряя магнитное поле, можно найти залежи полезных ископаемых, а может быть какие-нибудь инопланетные машины, спрятанные под землей.

Обычный компас и карта не помогут нам в этом, но на плате микроконтроллера micro:bit имеется встроенный магнитометр. Он способен измерять напряженность магнитного поля по трем осям.

На базе этого магнитометра вы можете сделать компас, или прибор для исследования магнитных полей, создаваемых, например, постоянными магнитами или соленоидами, для обнаружения металла или скрытой проводки.

Микросхема магнитометра находится в том же месте, что и акселерометр (рис. 6.1).

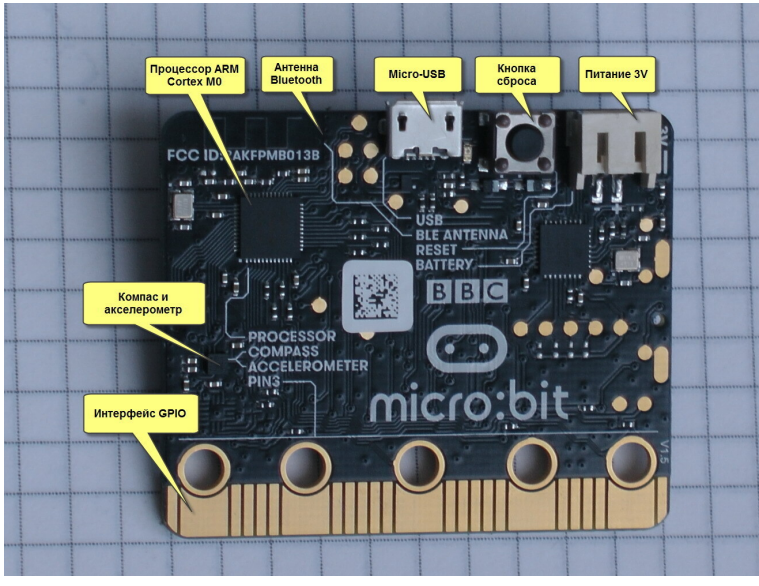


Рис. 6.1. Расположение магнитометра на плате micro:bit

Магнитометр micro:bit способен измерять интенсивность магнитного поля по трем пространственным осям (x, y, z). Он возвращает значения в микротеслах (мкТл).

Перед использованием необходимо выполнить калибровку магнитометра. К сожалению, эта процедура довольно длительная – вам нужно поворачивать и опрокидывать плату микроконтроллера до тех пор, пока на мониторе не будут гореть все светодиоды. Хотя калибровку можно запустить яв-

ным образом, она будет запущена автоматически в любом случае, если вы добавите в программу блоки магнитометра.

Обнаружение магнита

Давайте начнем изучение магнитометра с изготовления прибора для поиска магнитов и залежей железа. На рис. 6.2. показана программа, которая пригодится нам для такого прибора.

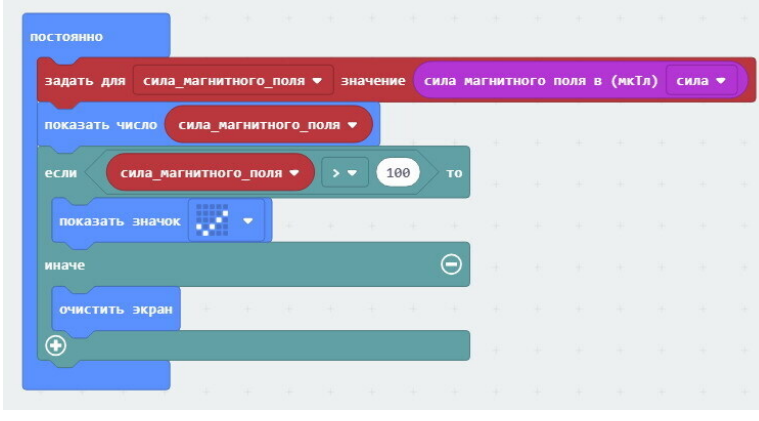


Рис. 6.2. Программа обнаружения магнитов

Код программы вы можете загрузить из файла BoxRover/ ch06/ microbit-Детектор-магнитного-поля.hex (файл находится в архиве на сайте автора книги http://frolov-lib.ru/ books/boxrover/).

Эта программа записывает в созданную нами переменную

сила_магнитного_поля значение интенсивности магнитного поля, полученное от блока **сила магнитного поля в (мкТл)** с параметром **сила**. Этот блок вы найдете в палитре **Ввод еще**.

Блок **сила магнитного поля в (мкТл)** в зависимости от выбранного параметра может возвращать значение силы магнитного поля по осям (x, y, z), или интенсивность магнитного поля, если задан параметр **сила**.

В бесконечном цикле наша программа показывает на экране micro:bit численное значение интенсивности магнитного поля в микротеслах. Если интенсивность превышает 100 мкТл, то на экране отображается значок галочки, если она меньше этого значения – экран стирается при помощи блока **очистить экран**.

Сразу после запуска программы на мониторе micro:bit в режиме бегущей строки появится надпись TILT TO FILL SCREEN, что можно перевести как «наклоняйте, чтобы заполнить экран», а затем – мигающая точка в центре экрана. Это означает, что нужно выполнить калибровку магнитометра.

Наберитесь терпения, поворачивайте и опрокидывайте плату микроконтроллера до тех пор, пока не загорятся все светодиоды на его экране. Если в процессе калибровки снова появится упомянутая выше надпись, дождитесь когда она будет полностью показана, а затем продолжите повороты и опрокидывания платы micro:bit.

Постарайтесь по возможность проводить калибровку вдали от магнитов и массивных металлических предметов, чтобы избежать их влияния на этот процесс.

Как только калибровка будет завершена, на экране micro:bit появится текущее значение интенсивности магнитного поля. Оказалось, что у автора этой книги на столе это значение равно примерно 42-44 мкТл.

Теперь возьмите любой магнит и медленно подносите его к плате micro:bit. Вы будете фиксировать увеличение значения интенсивности магнитного поля, измеренного магнитометром.

Если поднести магнит достаточно близко к микроконтроллеру, так что интенсивность магнитного поля превысит 100 мкТл, то на экране вслед за цифровым значением будет показан символ галочки.

Когда вы уберете магнит подальше, ваш micro:bit будет фиксировать обычное значение магнитного поля, типичное для вашей обстановки.

Проверьте, как влияют на показания вашего прибора магнитики с холодильника и другие магниты, которые вы найдете у себя дома.

Если поднести к плате micro:bit очень сильный неодимовый магнит, то прибор «намагнитится» и какое-то время будет показывать завышенное значение магнитного поля даже после того, как вы уберете магнит. Поэтому лучше не экспериментировать со слишком сильными магнитами.

Вы можете попробовать подносить к магнетометру различные металлические предметы, провода и катушки провода, по которым идет ток. Следите за тем, как это влияет на показания вашего магнитометра.

Делаем из micro:bit компас

При помощи магнитометра, встроенного в micro:bit, можно сделать программный компас. В палитре **Ввод** есть блок компасный **курс**, который возвращает значение от 0 до 359 градусов. Значение 0 соответствует направлению на север, значение 180 – направлению на юг.

Простейшая программа компаса показана на рис. 6.3 (файл microbit-Компас.hex).

Конец ознакомительного фрагмента.