



ВВЕДЕНИЕ В ТЕХНОЛОГИЮ INTEL® EDISON КНИГА НОВИЧКА

Или освоение образовательного набора по электронике и программированию. Набор Роботов™ A1 от Maxxlife Robot®



РАСКРЫВАЕМ ТАЛАНТЫ В КАЖДОМ

Более 10 лет корпорация Intel развивает целый спектр образовательных программ в России и странах СНГ. Портфолио инициатив корпорации включает проекты, направленные на профессиональное развитие учителей, поддержку научно-технического образования, развитие технического предпринимательства и инноваций.

С 2014 года компания объединила свои социально значимые программы под единым названием «Раскрываем таланты в каждом». Адаптированные в соответствии с локальными потребностями образовательные и социальные инициативы Intel следуют единой цели - раскрыть потенциал каждого из участников, предоставляя необходимые инструменты и возможности для успешного развития в современной инновационной экономике.

Основными задачами программы являются развитие и поддержка молодого поколения инженеров, изобретателей и инноваторов, которые будут разрабатывать новые технологичные продукты в таких новых сегодня областях, как интернет вещей, носимая электроника, большие данные, а в будущем - определять и создавать новые тренды в индустрии. Сегодня самые важные прорывные разработки ведутся на стыке наук, например информатики и лингвистики (распознавание речи), информатики и биологии (биоинформатика, персональная медицина), информатики и механики (роботы) и т. д., поэтому крайне важно вовлекать школьников и студентов в междисциплинарные, разнонаправленные проекты, где информатика будет одними из основных компонентов, тесно связанным с другими дисциплинами. Ярким примером такой деятельности для школьников является робототехника – это интересно детям, это отвечает духу времени и запросу будущего, используется для практически значимых отраслей науки и промышленности. Задачи могут ставиться преподавателями с учебной целью или придумываться самими школьниками, в этом случае проявляются уже творчество и изобретательство (мейкерство), создается новый продукт.

Это издание призвано помочь учителям и педагогам дополнительного образования войти в мир робототехники, увлечься новым направлением и увлечь за собой ребят.

Желаем успехов в захватывающем мире мейкерства, программирования и изобретательства!

СОДЕРЖАНИЕ

Раскрываем таланты в каждом	3
Содержание	4
Авторское право	7
Об авторе	8
От автора	12
Пролог	13
Глава 1. Введение в «Набор роботов A1»	15
1.1 Инструкция по работе с Книгой новичка по Intel® Edison	15
1.2 Комплектация набора A1	17
Глава 2. Описание компонентов набора	18
Глава 3. Освоение Стенда новичка A1	24
Глава 4. Микрокомпьютер Intel® Edison	27
Глава 5. Установка драйверов для Intel® Edison на компьютер	29
5.1. Вариант № 1 ОС Windows	29
5.1.1. Инструкция № 1 «Установка драйверов для Intel® Edison на Windows x32»	29
5.1.2. Инструкция № 2 «Установка драйверов для Intel® Edison на Windows x64»	34
5.2. Вариант № 2 ОС Mac OS X	39
Глава 6. Среда разработки программ Arduino IDE	40
Глава 7. Краткая теория по программированию в среде Arduino IDE	43
7.1. Структура программы	43
7.2. Синтаксис	44
7.3. Переменные	44
7.4. Арифметические операторы	45
7.5. Операторы сравнения	46
7.6. Управляющие операторы	46
7.7. Функции	49
7.8. Отладка программ в Serial	54

Глава 8. Полезная информация перед началом работы	57
8.1. Памятка по работе с набором и платой	57
8.2. Схема выводов и основных разъемов платы Intel® Edison	59
Глава 9. Создание схем и их программирование	61
Схемы уровня сложности А	61
Схема 1. Моргание встроенным светодиодом	63
Схема 2. Постоянное свечение светодиода	66
Схема 3. Моргание внешним светодиодом	68
Схема 4. Прототип «умного» дома на основе светодиода и фоторезистора	71
Схема 5. Включение и выключение светодиода с помощью кнопки	75
Схема 5.1. Включение и выключение светодиода с помощью двух кнопок	77
Схема 6. Светорегулятор на основе потенциометра	79
Схема 7. Синтезатор звука на основе пьезоэлектрического звонка	81
Схема 8. Температурный датчик со световой сигнализацией	83
Схема 9. Создание музыкального инструмента	85
Схема 10. Многоцветный RGB-светодиод	87
Схема 11. Игра «Кнопочные гонки»	89
Схема 12. Включение и выключение светодиода с помощью веб-сервера	91
Схема 13. Электромотор, транзистор и диод	93
Схема 14. Сервопривод и управление им	96
Схема 14.1. Подключение сервопривода с дополнительным источником питания	99
Глава 10. Освоение операционной системы Yocto Linux на Intel® Edison	100
10.1. Обновление операционной системы Intel® Edison	101
10.1.1. Вариант № 1 ОС Windows 101	
10.1.2. Установка Putty	102
10.1.3. Инструкция № 3 «Обновление Intel® Edison для Windows x64 (с помощью Intel® Software Setup Assistant)»	102
10.1.4. Инструкция № 4 «Обновление Intel® Edison для Windows x32 x64 (вручную)»	105
10.1.5. Форматирование встроенного накопителя Intel® Edison из FAT в FAT32	106
10.1.6. Вариант № 2 ОС Mac OS X	109

10.2. Подключение Intel® Edison к Wi-Fi	110
Глава 11. Интернет вещей и Intel® Edison. Работа с облаком Intel® IoT Dashboard	113
11.1. Введение в IoT (Internet of Things) и облачный сервис Intel® IoT Analytics	113
11.2. Создание учетной записи на портале Intel® IoT Analytics	116
11.3. Настройка Intel® Edison и регистрация на портале Intel® IoT Analytics	118
11.4. Установка библиотек IoTKit для Arduino IDE	120
11.5. Схемы уровня сложности В	121
Схема В1. Управление светодиодом с помощью облака Intel® IoT Dashboard	121
Схема В2. Отправка данных с датчика температуры с помощью облака Intel® IoT Dashboard	124
Глава 12. Подключение внешних USB-устройств к Intel® Edison	128
Схема С1. Подключение веб-камеры к Intel® Edison	129
Потоковое вещание видео (видеостриминг)	129
Глава 13. Ошибки Intel® Edison и способы их устранения	137
13.1. Скетч не загружается на плату	137
13.2. Arduino IDE не видит нужный COM-порт	137
13.3. При загрузке скетча Arduino IDE пишет Transfer incomplete	138
13.4. Скетч долго компилируется в Arduino IDE	138
13.5. Putty не открывает нужный COM-порт	138
Глава 14. Классные проекты для твоего вдохновения	139
Эпилог	140
О корпорации Intel	141

АВТОРСКОЕ ПРАВО

Введение в технологию Intel Edison®, или Освоение образовательного набора по электронике и программированию

Набор роботов™ A1 от Maxxlife Robot®

Первое издание. Эта книга вышла при поддержке корпорации Intel

Copyright © 2015 Максим Массальский

ОБ АВТОРЕ

Здравствуй, дорогой читатель!

Меня зовут Максим Массальский, и я являюсь автором этой книги. Лично я считаю, что она никогда бы не увидела свет, если бы не мое случайное знакомство с сотрудником корпорации Intel® Карлосом Монтесиносом (Carlos Montesinos), который подарил мне плату Intel® Edison на выставке Maker Faire, проходившей в 2014 году в Риме. Огромное спасибо ему за это. Благодаря такому подарку я познакомился со многими сотрудниками и подразделениями Intel® и смог при поддержке Intel® выпустить данную книгу.

Сам я из Минска (Республика Беларусь). Предлагаю перенестись в мое прошлое, чтобы узнать обо мне немного больше. В робототехнику я пришел из технического творчества. Помню, как в пятом классе записался в кружок по авиамоделированию и с превеликим удовольствием на него ходил. Там мы собирали кордовые модели самолетов.

Затем я увлекся автомоделированием и стал создавать свои собственные конструкции автомоделей. В этом деле я добился больших высот, стал пятикратным чемпионом Республики Беларусь по автомоделному спорту в классе Багги 8Д, однако главным моим увлечением стала робототехника.

Впервые с роботами я познакомился в своей Гимназии № 5 г. Минска, когда там появились конструкторы роботов одной известной марки. В гимназии я и начал заниматься с конструкторами после уроков. Так начался мой путь в робототехнику. После уроков я вместе со своим учителем информатики долгое время собирал роботов и программировал их. Вскоре к нашей небольшой команде стали подтягиваться и другие ребята. Так у нас появился школьный кружок по робототехнике. Хочу поблагодарить свою любимую гимназию! Помню, как я в 2012 году занял третье место на соревнованиях Roborace по гоночным роботам среди студентов вузов Беларуси. Я был

там единственный школьник одиннадцатого класса, чем до сих пор очень горжусь! С готовыми конструкторами я получил базовые навыки, но понимал, что это тупиковый путь и надо двигаться дальше, в сторону создания своих собственных систем и механизмов.

После окончания гимназии я поступил в Белорусский национальный технический университет. Там я также хотел изучать и развивать робототехнику, но столкнулся с тем, что на нашем машиностроительном факультете на тот момент (2012 год) роботов не было. Пришлось всему учиться самому. Огромное спасибо моему отцу, что он в то время помог мне найти человека, профессионала по робототехнике, которому было интересно передать часть своих знаний в данной области мне. Его зовут Павел Яцевич. Так, в одном из помещений ДОСААФ г. Минска я стал учиться новому и впитывать информацию, которую мне сообщал Павел. Мы работали тогда как мастер и подмастерье. Это было трудное время, когда я постоянно узнавал что-то новое, читал много технической литературы и применял теорию на практике.

Я даже назвал наше место работы на калифорнийский манер – Green Valley. Кстати, наше знакомство с Павлом произошло в ноябре 2013 года при очень интересных обстоятельствах. Перед этим я купил набор робота в одном китайском магазине. И у меня никак не получалось запустить плату по управлению двигателями. Мой отец смог найти Павла на форуме и предложил ему обучить меня основам. Спасибо китайскому драйверу, который пришел ко мне с браком! Так мы работали около года и сделали вместе пару интересных проектов. Это балансирующий робот и беспилотный гоночный робот-автомобиль Maxxlife Robot I, он же Maxxlife Robot Rover, Maxxlife Robot NeXT. С этим проектом я побывал на многочисленных выставках в Минске, выиграл немалое число конкурсов и занял много призовых мест. Благодаря этому роботу нашел отличную работу, съездил в Рим на Maker Faire-2014, в Москву на Robofest-2015, где занял второе место в классе Roboracers. Он и сейчас в рабочем состоянии. А почитать про него можно здесь: <http://maxxlife.by/maxxliferr.html>.

После года работы наша миниатюрная команда поняла, что пора расширяться. Наша Green Valley переродилась и

сменила место дислокации. Так и появилась первая в Беларуси лаборатория по интернету вещей и робототехнике - «Кухня роботов» (vk.com/cookrobot). Теперь наша команда работает в просторном помещении, которое было выделено ДОСААФ. К нам постепенно стали приходить новые люди. А я после года интенсивной работы смог сам вести и разрабатывать проекты, продолжая сотрудничать с Павлом, т. к. это сокращало время работы над проектами. В нашей лаборатории «Кухня роботов» увидел свет проект колесного робота SmartMan, который, используя алгоритмы машинного зрения, и камеру, мог самостоятельно передвигаться. Интересный проект! Жаль, что я не придумал ему применение и не стал развивать дальше. В тот период я разработал двух образовательных роботов: StartBot A (<http://maxxlife.by/startbota.html>) и StartBot X (<http://maxxlife.by/startbotx.html>).

Время летело, а я все не знал, в какой проект установить Intel® Edison. И как всегда, мне на помощь пришел случай. Если заметили, то каждый случай тянет за собой другой случай. Удивительно, не правда ли? Я узнал про семинар по камере Intel® RealSense™, который проходил в Санкт-Петербурге 30 мая 2015 года. Очень хотелось побывать на этом мероприятии, но у меня не было финансовой возможности это сделать. Тогда я обратился к директору компании «КомплИТех» Алексею Лагойко с просьбой о материальной помощи для поездки (меня пригласили на работу в эту компанию, и я проработал там один год). Он ответил согласием, часто мне помогал. Спасибо ему огромное за это :)

На семинаре я удивительным образом познакомился с Романом, программистом из Минска, который приехал туда, чтобы всех удивить. Ему это удалось, он занял первое место со своим проектом очков виртуальной реальности, а я завязал очень важное знакомство, о котором еще не догадывался. Спустя неделю после семинара Роман мне позвонил и предложил сделать совместный проект для хакатона Intel® IoT RoadShow, который должен был проходить опять в Петербурге, но уже 27 июня. Я согласился. Идея проекта заключалась в том, чтобы сделать робота, который бы мог управляться прямо из виртуальной реальности! Я сконструировал робота на базе Intel® Edison, а Роман написал ПО для очков вир-

туальной реальности и камеры Intel® RealSense™. Наш проект мы назвали SenseDrive™ – и заняли второе место на этом хакатоне! Почитать про наш проект можно здесь: <https://habrahabr.ru/company/Intel®/blog/261747/>.

Со временем я решил коммерциализировать свои разработки. Так появился в Беларуси интернет-магазин для инновационного творчества «М-Робот™». Товары – это «Набор роботов™ А1», а также готовые комплекты образовательных колесных роботов «Платформа™», чтобы у новичков не было трудностей при создании собственных прототипов или в постижении новых дисциплин.

Сейчас наша «Кухня роботов» занимается разработкой образовательных наборов по интернету вещей и робототехнике, дальнейшим развитием проекта SenseDrive™ (мы добавили нашему роботу биомеханическую руку), программированием системы машинного зрения для БПЛА; создает дизайн «умной» одежды, а также конструирует беспилотный летательный аппарат вертикального взлета и посадки с возможностью доставки грузов.

Найти меня можно здесь: vk.com/maxxlife и facebook.com/teammaxxlife.

Огромное спасибо моей семье за то, что всегда была со мной при написании этой книги.

Спасибо моему отцу за то, что поддерживал меня и мои начинания.

Спасибо корпорации Intel® и ее сотрудникам в разных странах за то, что позволили этой книге появиться на свет.

Отдельное спасибо моей команде, без которой я бы не смог уйти так далеко.

Спасибо всем, кто прошел со мной этот путь. Было классно!

Мне 20 лет, и думаю, что все еще впереди. И... побольше общайтесь с людьми!

ОТ АВТОРА

Вас приветствует команда разработчиков роботов и мехатронных устройств «Maxxlife Robot».

Наша команда при поддержке корпорации Intel® обеспечивает все условия для развития сообщества по робототехнике и устройствам «сделай сам». Наша цель – предоставить любому желающему возможность воплотить свои мечты в реальность. Специально для этого мы создали начальный образовательный набор по электронике и программированию «Набор роботов™ A1», к которому и прилагается эта книга. Творите с удовольствием!

Мы надеемся, что создание электронных устройств и их программирование станет для вас увлекательным и интересным занятием. Расширяя свои горизонты, вы сможете по-другому взглянуть на окружающий мир. На этом пути неизбежно будут встречаться трудности однако не стоит отчаиваться. Человек, который пишет сейчас эти слова, тоже сталкивался с препятствиями и сложностями, но благодаря упорству и терпению, смог стать знатоком своего дела и хорошо понимает как сложно начинать изучать новые направления. В этой книге сделано все возможное, чтобы облегчить ваш путь к знаниям.

У нас существует поддержка клиентов. Связаться с нами можно по почте maxxliferobot@gmail.com либо создать тему на нашем форуме forum.mrobot.by и задать любой интересующий вас вопрос по использованию нашего продукта или решению какой-либо проблемы, связанной с «Набором роботов™ A1».

На сайте Intel® есть англоязычный форум с большим количеством высококвалифицированных специалистов компании Intel®, где вы можете задавать вопросы, рассказывать о своих изобретениях с окружающими и узнавать, какими проектами занимаются другие.

ПРОЛОГ

В первую очередь я хочу ответить на вопрос: «Кому предназначается эта книга?»

Эту книгу я писал для тех, кто хочет освоить высокие технологии, но не знает, с чего начать. Ведь что может быть лучше, чем сделать свой дом «умным»? А может, создать своего собственного робота или устройство по слежению за дверцей холодильника? Или же кто-то вынашивает идею нового прибора, которому суждено перевернуть мир, но не знает, с чего начать?

Хочу сразу предупредить, что после прочтения вы не станете великими программистами или электронщиками, но поймете самое важное – то, как им стать и что нужно для этого делать. Это непросто, поверьте мне, успех в любом деле добывается по крупицам, которые вместе составляют пирамиду на пути к вашей цели.

Книга предназначается всем тем, кто не боится учиться новому и получать современные знания. Тем, кто не боится трудностей и не любит сдаваться.

Для того чтобы сделать успешным освоение электронных схем и чуть облегчить изучение материала, был создан «Набор роботов А1», в котором есть все для удачного старта. Теперь не надо бегать в магазин и искать нужные детали. Все под рукой.

Предлагаю перейти на «ты». Так мне будет проще взаимодействовать с тобой, дорогой читатель. На протяжении всей книги я буду рассказывать и показывать, что именно надо делать в той или иной главе либо схеме, куда подключать компоненты; решать возникшие в процессе работы вопросы, чтобы ты смог успешно освоить эту книгу. Я пишу этот текст так, чтобы с минимальными усилиями можно было научиться создавать электронные устройства различных уровней сложности.

1

**ТЕОРИЯ
И ПОДГОТОВКА**

ГЛАВА 1

ВВЕДЕНИЕ В «НАБОР РОБОТОВ A1»

1.1. Инструкция по работе с Книгой новичка по Intel® Edison

Практически к каждой главе книги, я буду давать ссылку на файл, где описаны все неточности и правки, которые могут со временем возникнуть. К сожалению, за скоростью информационных технологий печатному станку не угнаться. Все стабильные исходники программ, которые были использованы при создании книги, находятся по ссылке <http://tiny.cc/a1-source>. При возникновении трудностей попробуй работать с программами из этого источника.

Внимательно изучи оглавление книги. Она состоит из разделов, глав и пунктов. Первый раздел посвящен освоению электрических компонентов, изучению платы Intel® Edison и ее подготовке к работе, знакомству со средой программирования Arduino IDE, а также теории по программированию в Arduino IDE. Второй раздел посвящен применению полученных знаний на практике, сборке электрических схем и программированию Intel® Edison. Во втором разделе речь пойдет о четырех видах схем. Все схемы в книге разделены на уровни сложности: **A, A+, B, C**.

A – это схемы начального уровня, для создания которых требуется минимальное количество знаний, полученных из книги. Нельзя сказать, что все схемы уровня A очень простые, и ты сможешь их сразу собрать, однако многие из них рассчитаны на человека, который до этого момента совсем не имел дела с электроникой.

A+ – это схемы более высокого уровня, для создания которых требуются компоненты, которые не входят в стандартную комплектацию набора, – их надо приобретать отдельно. В этих схемах присутствуют сложные устройства и (или) программы, поэтому для полного понимания работы таких схем, потребуются дополнительные знания в области электроники,

механики и программирования, которые можно получить, набрав адреса ссылок на электронные информационные ресурсы, приведенные в тексте схем A+.

В – это схемы, в которых используются не только Arduino-программы (скетчи), но и элементы операционной системы Yocto Linux и веб-сервисы. Для успешной сборки схем необходимо внимательно изучать предыдущий материал.

С – это схемы высокого уровня, и приступать к ним можно только тогда, когда будут успешно освоены все предыдущие схемы. В **С**-схемах программы напрямую выполняются в операционной системе Yocto Linux. Для создания таких схем требуются дополнительные элементы, поэтому внимательно читай, что необходимо для работы.

В тексте довольно много ссылок на информационные ресурсы – очень советую пользоваться ими.

Таким шрифтом будут набраны команды, которые необходимо вводить в системе, элементы кода программы, а также имена клавиш, которые надо нажимать. Например: введи команду `reboot ota`, выполни операцию `val=analogRead(0);`, нажми на клавишу Enter. Таким способом будут выделены ссылки на интернет-ресурсы. Например, intel.com, mrobot.by.

Чтение книги следует начинать с первого раздела, не пропуская далее ни одной главы или пункта. Ведь на протяжении всей книги дается информация, которая обязательно потребуется в дальнейшем.

И помни: нетерпение и невнимательность в любом деле ведут к неудачам, а в электронике – к непоправимым последствиям.

1.2. Комплектация набора A1

1. Стенд новичка A1 (далее также Стенд).
2. Микрокомпьютер Intel® Edison с Arduino-платой расширения (далее – плата, главная плата, основная плата). Микрокомпьютер заранее установлен на Стенд новичка. (What will you make?– гласит надпись на микрокомпьютере. «А что сделаешь ты?» – можешь задать себе такой вопрос. Я отвечу: «Всё, что пожелаешь!»)
3. В коробке компонентов A1 (далее – Коробка A1) Стенда находятся следующие детали:

- 10 штук резисторов с сопротивлением 10 кОм.
- 10 штук резисторов с сопротивлением 220 Ом.
- 1 фоторезистор 10 кОм.
- 1 термистор 10 кОм.
- 4 тактовые кнопки.
- 1 пьезодинамик.
- 20 штук проводов «папа-папа».
- 1 многоцветный светодиод.
- 2 красных светодиода.
- 2 зеленых светодиода.
- 1 желтый светодиод.
- 1 синий светодиод.
- 1 пластиковая крепежная перемычка.
- 2 винта.
- 2 пластиковые гайки.

Комплектация «Набора роботов A1» может незначительно изменяться.

4. Под Коробкой A1 находится 1 микро-USB-кабель.
5. Одна беспаячная макетная плата (приклеена на крышке Коробки A1).

Все эти компоненты позволят тебе собрать 12 А-схем, 2 В-схемы. Для остальных схем потребуются дополнительные детали, которые ты сможешь докупить отдельно.

ГЛАВА 2

ОПИСАНИЕ КОМПОНЕНТОВ НАБОРА

Набор создан таким образом, чтобы с ним мог работать даже человек далекий от электроники. Здесь представлены краткие сведения широко распространенных электронных компонентах и приборах, а также о признаках, по которым их можно определить. В любом случае, если тебе непонятно, как работает один из компонентов, то в интернете полно информации по каждому из них. Дополнительная информация и возможные правки к главе доступны по ссылке <http://tiny.cc/a1-edit-c2>.

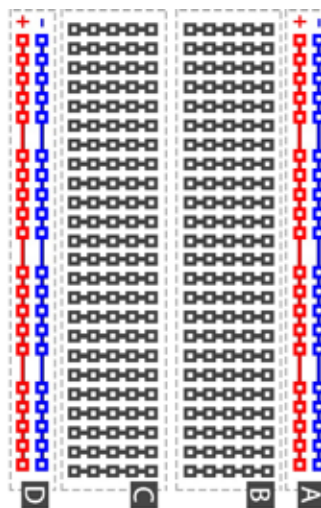
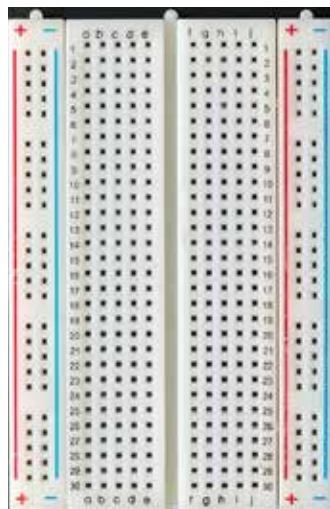
Макетная плата

Зачем она нужна. Беспаячная макетная плата служит для быстрого прототипирования электронных устройств без применения паяльника. Это позволяет избежать ошибок при конструировании схем, также быстро внести изменения в прототип разрабатываемого устройства и тут же проверить результат. Макетная плата, безусловно, экономит время и является очень полезной.

Как определить: Выглядит как прямоугольная пластина с большим количеством отверстий на одной из сторон.

Важные особенности. С беспаячной макетной платой необходимо понимать как она устроена.

В отверстия вставляются необходимые электронные компоненты. Однако каждое отверстие не является изолированным от другого, есть определенная система. Как видно на рисунке слева, если поставить плату вертикально, отверстия в вертикальных столбцах соединены между собой, а сами столбцы не соединены. Обычно такие выводы используются для подачи питания, когда плюс и минус не должны быть замкнуты без нагрузки (А). Теперь в середине платы тоже есть отверстия (Б). Они соединены горизонтально, т. е. все отверстия в строке объединены (замкнуты), а сами строки друг от друга изолированы (см. рисунок слева).





Светодиод (LED)

Что он делает. Светодиод излучает свет, если пропустить через него небольшой ток. Другими словами, это надежная и энергоэффективная лампочка. Имеет полярность.

Как определить. Обычно выглядит как полупрозрачный закругленный цилиндр с двумя выводами.

Количество выводов. Два (длинная ножка – это плюс, короткая – минус).

Важные особенности. Необходима установка резистора, ограничивающего ток.



Резистор

Что он делает. Резистор ограничивает количество электрического тока, который может протекать по электрической цепи. Другими словами, это своеобразный искусственный барьер для тока, где часть энергии, которая не прошла сквозь резистор, превращается в тепло.

Как определить. Выглядит как маленький цилиндр с нанесенными на него по кругу полосками. Из торцов цилиндра торчат тоненькие металлические проволочки (выводы). Для определения сопротивления резистора используется цветовая кодировка.

Количество выводов. Два.

Важные особенности. При выборе резистора нужного номинала следует внимательно смотреть на его цветовую кодировку, т. к. можно легко ошибиться в определении цветов, нанесенных на резистор. Полярности не имеет, поэтому, как его подключать к схеме, разницы нет.

Таблица цветовых кодировок резисторов

Если присмотреться, то на резисторе четыре полоски и расстояние между тремя из них одинаковое, а четвертая стоит чуть дальше. Четвертая полоска показывает погрешность (на рисунке ниже она крайняя справа) резистора, а остальные – цифру и множитель. Благодаря этому рисунку тебе будет проще работать с резисторами и искать резистор нужного номинала, если это будет необходимо.

	1-я цифра	2-я цифра	множитель	погрешность
черный	-	0	x 1	серебряный 10%
коричневый	1	1	x 10	золотой 5%
красный	2	2	x 100	красный 2%
оранжевый	3	3	x 1.000	коричневый 1%
желтый	4	4	x 10.000	зеленый 0.5%
зеленый	5	5	x 100.000	фиолетовый 0,1%
синий	6	6	x 1.000.000	
фиолетовый	7	7	золотой x 0.1	
серый	8	8		
белый	9	9		

Например, тебе нужен резистор в 560 Ом. Смотрим по таблице. Цифра «5» соответствует зеленой полоске (первый столбец), цифра «6» синей (второй столбец). Получается 56. Но тебе нужен резистор в 560 Ом. Поэтому последняя полоска – это полоска-множитель, она должна быть равна 10. Такой цифре соответствует коричневая полоска (третий столбец). $56 * 10 = 560$. Нужный резистор найден. Если нужно определить по резистору сопротивление, надо просто цвет полосок, начиная с первой, сопоставить с рисунком выше.

Тактовая кнопка

Что она делает. Замыкает электрическую цепь, когда нажата.



Как определить. Выглядит как квадратика с кнопкой наверху и четырьмя короткими выводами.

Количество выводов. Четыре.

Важные особенности. Кнопка полностью квадратная. Два вывода на одной стороне и два на противоположной.

Пьезоэлектрический звонок

Что он делает. Пьезоэлектрический звонок создает звук, используя для этого пьезоэлектрический эффект.



Как определить. Выглядит как черный цилиндр диаметром около 10 мм с отверстием с одной стороны и двумя выводами с другой.

Количество выводов. Два.

Важные особенности. Высокий и противный звук. Имеет полярность. Плюс подписан на верхнем торце с отверстием, соответственно, неподписанный вывод – минус. Также более длинный вывод – это плюс, а, соответственно, короткий – минус.

Потенциометр (реостат)

Что он делает. Создает различное сопротивление в зависимости от положения ручки регулировки. Потенциометры используются в робототехнике как регуляторы различных параметров: громкости звука, напряжения, угла поворота сервопривода и т. п.



Как определить. Это устройство может иметь разный форм-фактор. Однако принцип у всех один: есть вал, который может поворачиваться на определенное количество градусов, и три вывода (в основном).

Количество выводов. Три.

Фоторезистор

Что он делает. Создает переменное сопротивление в зависимости от количества света, падающего на фоторезистор. Проще говоря, это «умный» резистор. Электронные компоненты, работающие по схожему принципу, стоят в смартфонах, чтобы определять степень освещенности экранов.

Как определить. Представляет собой маленькую «таблетку». Одна ее сторона белая, со змейкой; от противоположной отходят два вывода.

Количество выводов. Два.

Важные особенности. Для работы необходим делитель напряжения из резистора на 10 кОм и фоторезистора. Прочти про делитель напряжения здесь: <http://tiny.cc/a1-voltage-divider>. Полярности не имеет, поэтому, как его подключать к схеме, разницы нет.



Термистор



Что он делает. Создает переменное сопротивление в зависимости от температуры среды, в которой находится сенсор. Принцип действия схож с работой фоторезистора.

Как определить. Представляет собой маленькую «таблетку» с двумя выводами.

Количество выводов. Два.

Важные особенности. Для работы необходим делитель напряжения из резистора на 10 кОм и самого термистора. Прочти про делитель напряжения здесь: <http://tiny.cc/a1-voltage-divider>. Полярности не имеет, поэтому подключен к схеме может быть по-разному.

ДОПОЛНИТЕЛЬНЫЕ КОМПОНЕНТЫ, КОТОРЫЕ МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ В СХЕМАХ

Дополнительные компоненты приобретаются отдельно либо могут входить в «Набор роботов A1 расширенный» или «Набор роботов A1 для учебных заведений».

Электромотор



Что он делает. Преобразует электрическую энергию в механическую. Вал электромотора вращается, когда через него протекает электрический ток.

Как определить. Представляет собой металлический корпус с крутящимся выходным валом.

Количество выводов. Два.

Важные особенности. Электромотор постоянного тока, не имеет полярности. При изменении направления протекания тока, мотор только поменяет направление вращения.

Транзистор



Для чего он нужен. Транзистор необходим для преобразования или усиления электрических сигналов.

Как определить. По маркировке. Выпускается в различных корпусах.

Количество выводов. Три.

Важные особенности. Не путать выводы (база, коллектор, эмиттер). Для ограничения тока на транзистор часто используются резисторы.

Диод



Для чего он нужен. Электронный эквивалент однонаправленного клапана. Ток через диод может течь только в одну сторону.

Как определить. Обычно цилиндрической формы с выводами на противоположных сторонах (на одной стороне нанесена светлая полоска, определяющая полярность).

Количество выводов. 2

Важные особенности. Пропускает ток только в одном направлении. Ток будет течь, если сторона с полоской подключена к низкому потенциалу, а другая сторона к более высокому.

Сервопривод

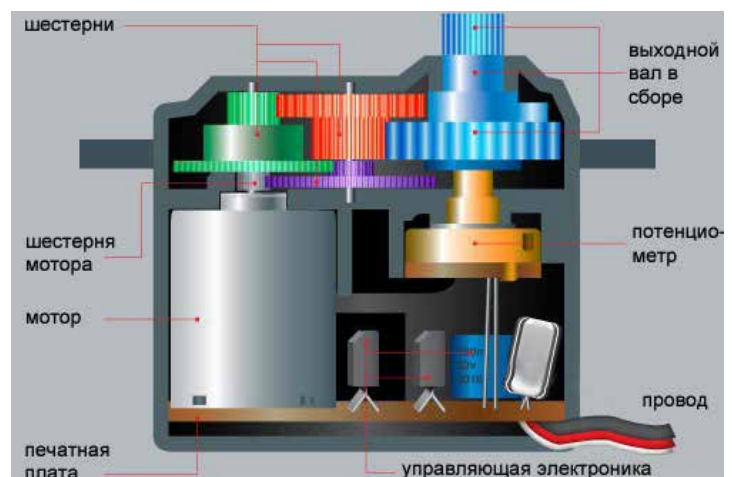


Что он делает. Так же как и электромотор, преобразует электрическую энергию в механическую. Однако сервопривод может удерживать заданное положение выходного вала.

Как определить. Сервоприводы бывают разных размеров и видов. Стандартные сервоприводы для использования в моделизме выглядят как прямоугольные корпуса с валом размером со спичечный коробок.

Количество выводов. 3 (Питание, земля, сигнал).

Важные особенности. Стандартный сервопривод управляется ШИМ-сигналом. Максимальный угол поворота вала обычно не более 180 градусов. Т. е. сделать полный оборот на 360 градусов вал сервопривода обычно не может.



ГЛАВА 3

ОСВОЕНИЕ СТЕНДА НОВИЧКА А1

Стенд служит для того, чтобы устанавливать на него компоненты и комфортно работать. А также для того, чтобы все лежало в одном месте и не терялось. Работать с книгой ты можешь и не имея данного приспособления. Дополнительная информация и возможные правки к главе доступны по ссылке <http://tiny.cc/a1-edit-c3>.

Набор А1 поставляется в уже собранном виде. В него входят микрокомпьютер Intel® Edison с Arduino-платой расширения, Коробка электрических А1, а для создания схем на крышке Коробки установлена безопасная макетная плата. Под Коробкой компонентов А1 лежит один микро-USB-провод для подключения Intel® Edison к компьютеру.

«Набор роботов А1». Вид сверху



«Набор роботов А1». Вид сверху со снятой крышкой

Верхняя крышка Коробки А1 имеет специальные выступы, благодаря которым можно легко ее подсоединить к основанию Стенда. В основании Стенда по периметру сделаны

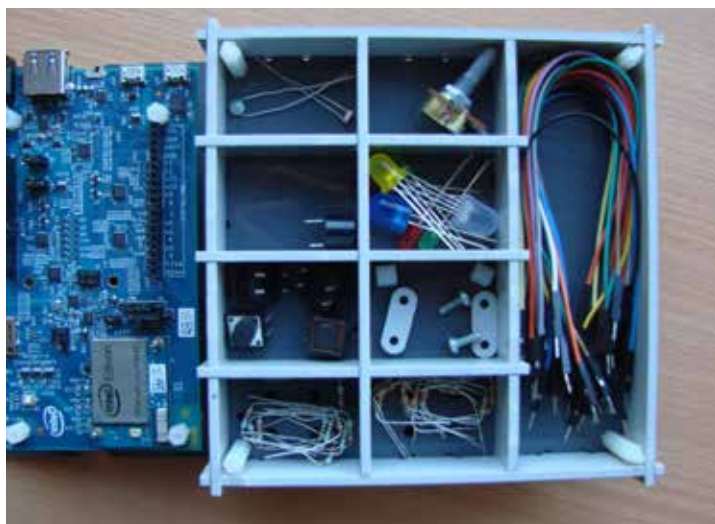


специальные вырезы-замки для крепления крышки Коробки и других дополнительных модулей.

Также верхняя крышка Коробки А1 имеет дополнительные отверстия, установив в которые пластиковые стойки можно поставить еще одну коробку и таким образом увеличить количество компонентов набора. Дополнительные пластиковые стойки и винты идут только в версии «Набор роботов А1» для образовательных учреждений, кружков и секций.

Коробка компонентов А1

В ней лежат необходимые детали для сборки всех схем из данной книги. В одном из отделений сложен крепеж (винты, перемычки, пластиковые гайки) для более надежного крепления крышки Коробки компонентов, если это необходимо. В самом длинном отделении лежат провода для коммутации схем.



Крепление верхней крышки Коробки А1 к Стенду

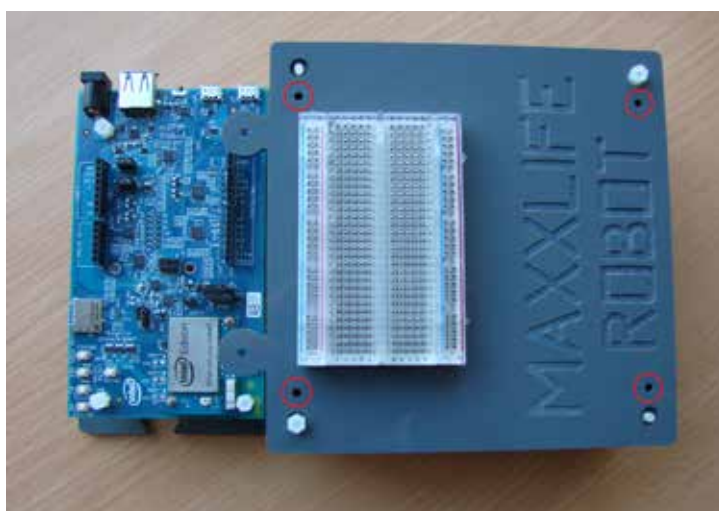
Крышка Коробки А1 сделана таким образом, чтобы удобно соединяться с нижней пластиной Стенда для комфортной работы. Сама крышка имеет «уши», которые устанавливаются в специальные вырезы-замки на Стенде. Для надежного соединения можно использовать крепежную перемычку, два винта и две пластиковые гайки. Один винт вставляется в отверстие на нижней пластине Стенда, второй – в отверстие на крышке



Коробки А1, сверху ставится крепежная перемычка, после этого пластиковые гайки заворачиваются на винты. Тем самым можно обеспечить надежное крепление крышки Коробки А1 к нижней пластине Стенда. Данным способом можно также крепить другие дополнительные модули и пластины, которые приобретаются отдельно.



Версия набора «Для учебных заведений» имеет дополнительные стойки и винты, которые крепятся в специальные отверстия на крышке Коробки компонентов (отмечены красным). Благодаря этим стойкам наборы А1 можно складывать друг на друга: шпильки стоек на крышке первого стенда войдут в отверстия на нижней пластине второго и т. д. Ученикам и преподавателям будет удобно складывать наборы и хранить их.



ГЛАВА 4

МИКРОКОМПЬЮТЕР INTEL® EDISON



На Стенде новичка ты видишь маленькую блестящую пластинку размером с SD-карту и надписью Intel® Edison. Это и есть микрокомпьютер, который работает под управлением операционной системы Linux Yocto.

Микрокомпьютер использует 22-нм технологию SoC (System on chip) от Intel®, которая включает двухъядерный процессор Intel® Atom на частоте 500 МГц, и 32-битный Intel® Quark-микроконтроллер, который работает на частоте 100 МГц. А это довольно большая вычислительная мощность для микрокомпьютера чуть больше стандартной SD-карты!

Intel® Edison поддерживает 40 GPIO, включает в себя 1 ГБ LPDDR3, 4 ГБ EMMC, и имеет двухдиапазонный Wi-Fi и BTLE, что делает его еще более привлекательным. По умолчанию плата поддерживает операционную систему для встраиваемых устройств Linux Yocto OS.

В твоем случае, микрокомпьютер установлен на Arduino-плату расширения. Эта плата была разработана для аппаратной совместимости с платформой Arduino. А также, чтобы больше пользователей Arduino переходили на работу с Intel® Edison без всяких проблем. Проще говоря, Intel® Edison – это компактный микрокомпьютер, который позволит расширить возможности твоих прототипов устройств в сфере высоких технологий и сделать их еще умнее и совершеннее!



Подробнее про устройство Intel® Edison ты можешь прочесть здесь: <http://tiny.cc/a1-about-edison>.

Убедись, что черный микропереключатель стоит в крайнем положении, как показано на фотографии. Микропереключатель должен быть сдвинут в сторону двух микро-USB-портов.

Теперь тебе надо установить драйвера для работы с платой. Переходим к следующей части книги. Дополнительная информация и возможные правки к главе доступны по ссылке: <http://tiny.cc/a1-edit-c4>.

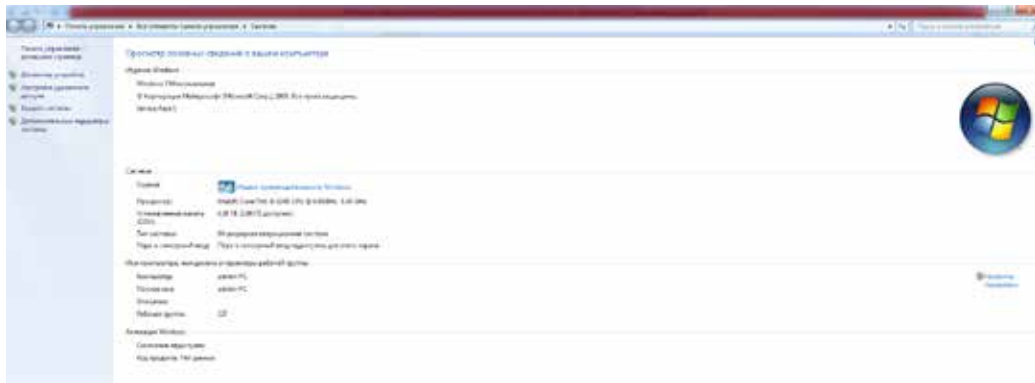
ГЛАВА 5

УСТАНОВКА ДРАЙВЕРОВ ДЛЯ INTEL® EDISON НА КОМПЬЮТЕР

5.1. Вариант № 1 ОС Windows

От разрядности твоей ОС зависти номер инструкции по установке драйверов. Чтобы узнать разрядность операционной системы Windows, тебе надо нажать сочетание клавиш Win и Pause Break. Появится такое окно или похожее. В разделе «Система» будет указан тип системы: 64-разрядная или 32-разрядная. В зависимости от того, какая у тебя установлена ОС, выбирай инструкцию ниже.

Дополнительная информация и возможные правки к главе доступны по ссылке: <http://tiny.cc/a1-edit-c5>.



5.1.1. Инструкция № 1 «Установка драйверов для Intel® Edison на Windows x32»

При возникновении трудностей оригинал инструкции на английском языке ты можешь открыть по ссылке: <http://tiny.cc/a1-edison-start-win-x32>.

Что необходимо:

1. Компьютер с 32-битной версией Windows 7, Windows 8.
2. Два микро-USB-кабеля (один есть в наборе, второй, скорее всего, ты сможешь найти дома).
3. Плата Intel® Edison.

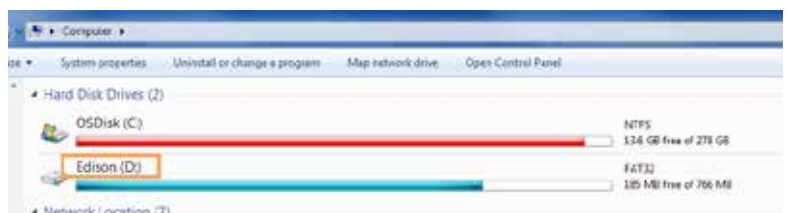
Эта последовательность шагов проверена на Windows 7x32. Установка драйверов тебе может показаться сложным и скучным занятием, но поверь мне, как только ты это сделаешь и начнешь создавать и программировать электронные схемы из этой книги, все потрясающие секреты откроются сами собой. А теперь за дело!

Шаг 1. Подключи два микро-USB-провода к Intel® Edison, а с другой стороны подсоедини их к двум USB-портам компьютера. Запомни, всегда сначала подключаешь центральный микро-USB-кабель, а потом крайний.

После подключения на плате Intel® Edison должен загореться зеленый светодиод. Если ты используешь ноутбук, то подсоедини его к зарядке, чтобы USB-порты выдавали максимально возможные напряжение и ток.



Шаг 2. После того как ты подключил Intel® Edison к компьютеру, тебе необходимо проверить, видно ли устройство как съемный диск. Оно появится через пару минут после подключения Intel® Edison к компьютеру.



Шаг 3. Установка USB-драйвера. Скачиваем установщик под названием Windows Standalone Driver 1.2.1 с сайта Intel® по ссылке: <http://tiny.cc/a1-edison-downloads> (необходимо прокрутить страницу сайта немного вниз).

Номер версии программы Windows Standalone Driver может быть другим, это не критично.

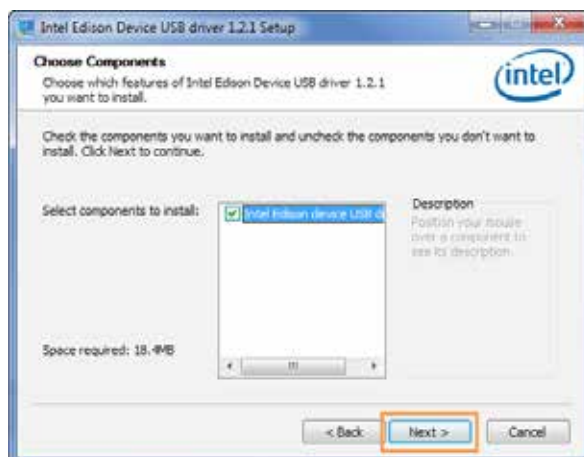
Шаг 4. Запусти загруженное приложение. Кликни Next.



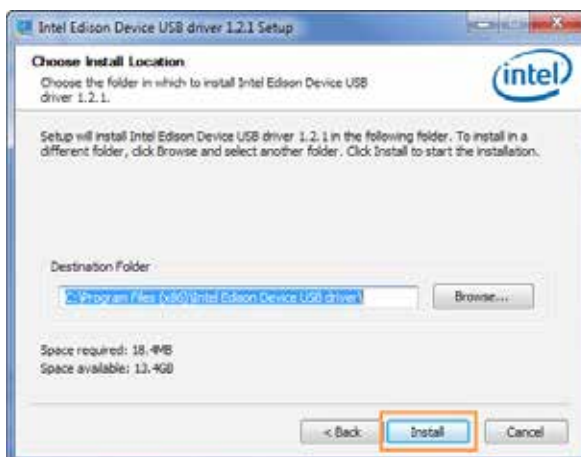
Шаг 5. Выбери I Agree.



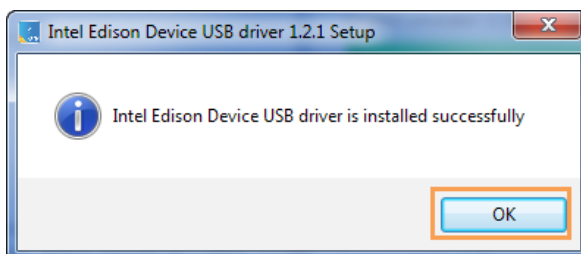
Шаг 6. Опять кликни Next.



Шаг 7. Выбери `Install`. Начнется установка драйверов. Это может длиться до 20 минут. Поэтому жди.



Шаг 8. Кликни `OK`.



Шаг 9. Кликни `Finish`.



Шаг 10. Теперь отсоедини два микро-USB-кабеля от платы Intel® Edison и подсоедини их обратно.

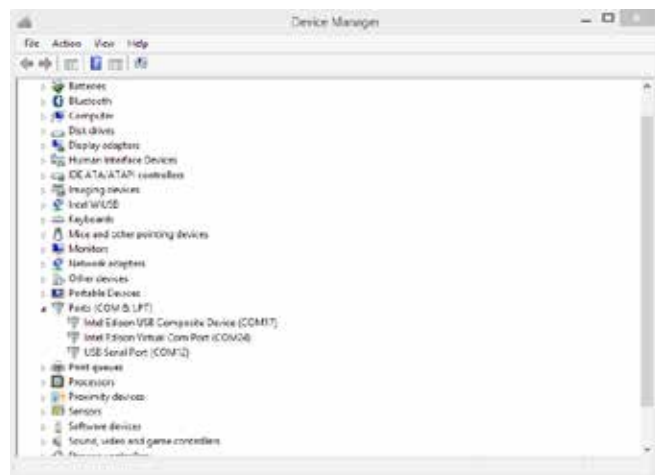
Шаг 11. Установка FTDI-драйверов. Скачай установочный файл по ссылке: <http://tiny.cc/a1-drivers-ftdi>. На странице сайта кликни по setup executable.



Шаг 12. Запусти программу от имени Администратора, кликнув правой кнопкой мыши по программе и выбрав «Запустить от имени Администратора».



Шаг 13. Следуй инструкциям по установке, после чего перезагрузи компьютер, не отсоединяя Intel® Edison.



Шаг 14. Проверочный. После того, как ты перезагрузил компьютер, проверь, правильно ли ты установил драйвера. Для этого зайти в «Диспетчер устройств».

В появившемся окне найди вкладку «Порты» (COM и LPT). Раскрой выпадающий список. Там должны быть два вида COM-портов со словами Intel Edison в названии (Intel Edison USB Composite Device, Intel Edison Virtual Com Port) и COM-порт USB Serial Port.

Возможна ситуация, при которой порт Intel Edison USB Composite Device не появится. Не обращай на это внимания. Главное, чтобы были два остальных порта. Если это так, переходи к следующим действиям, запомнив, под каким номером указан COM-порт Intel Edison Virtual Com Port. В моем случае это COM24. У тебя, скорее всего, будет другой. Также запомни номер, под которым находится USB Serial Port. На моем компьютере это COM12. Данный порт понадобится при работе с платой при помощи программы Putty, через терминал. Это для продвинутого пользователя, которым ты обязательно станешь, но чуть позже.

Если у тебя нет одного из нужных COM-портов (Intel Edison Virtual Com Port и USB Serial Port) или ни один порт не появился, попробуй отсоединить плату Intel® Edison от компьютера и подключить заново, подождав после повторного подключения 2–3 минуты. Если у тебя не появились два необходимых COM-порта, то пиши нам в службу поддержки!

Шаг 15. Установка Arduino IDE. Скачай установочный файл среды Arduino по ссылке: <http://tiny.cc/a1-arduino-download>. На странице сайта выбери Windows Installer. После чего открой его и следуй инструкциям по установке. Все пути установки программы оставь по умолчанию. После того как ты установишь программу, переходи к следующей части Книги новичка.

5.1.2. Инструкция № 2 «Установка драйверов для Intel® Edison на Windows x64»

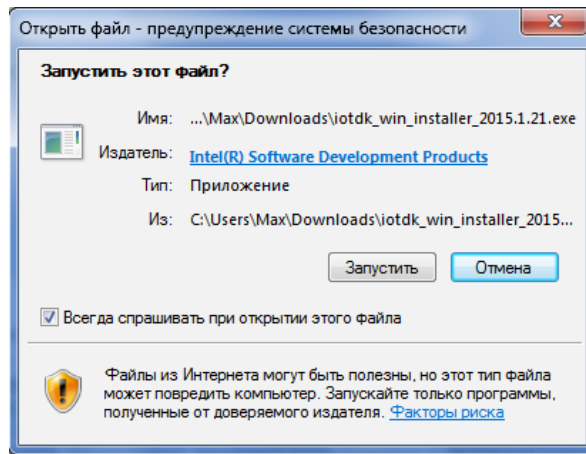
Что необходимо:

1. Компьютер с 64-битной версией Windows 7, 8, 8.1, 10.
2. Два микро-USB-кабеля.
3. Плата Intel® Edison.

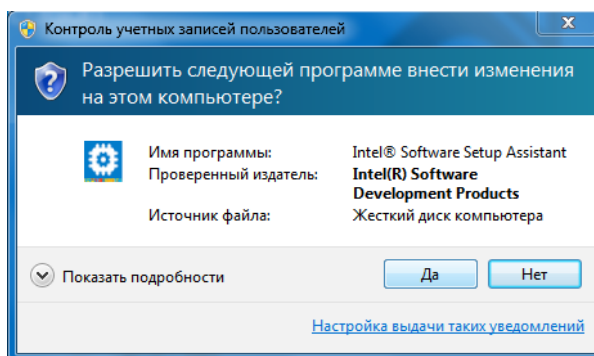
Такая последовательность шагов проверена на Windows 7 x64, и Windows 8.1 x64 и Windows 10 x64. Установка драйверов тебе может показаться сложным и скучным занятием, но поверь мне, как только ты это сделаешь и начнешь «играться» с нашим набором и его возможностями, все потрясающие секреты откроются сами собой. А теперь за дело!

Шаг 1. Скачай установщик под названием Windows 64-bit Integrated Installer с сайта Intel® по ссылке <http://tiny.cc/a1-edison-downloads> (необходимо прокрутить страницу сайта немного вниз).

Шаг 2. Убедись, что ты не подключил Intel® Edison к компьютеру. Открой загрузившееся приложение. Если выскочит окно «Предупреждение системы безопасности», то выбери Запустить.



Если откроется вкладка «Контроль учетных записей пользователей», то выбираем Да.



После этого появится окно загрузки приложения.



Шаг 3. После загрузки приложения откроется окно приветствия. Выбираем Next и переходим к следующему шагу.



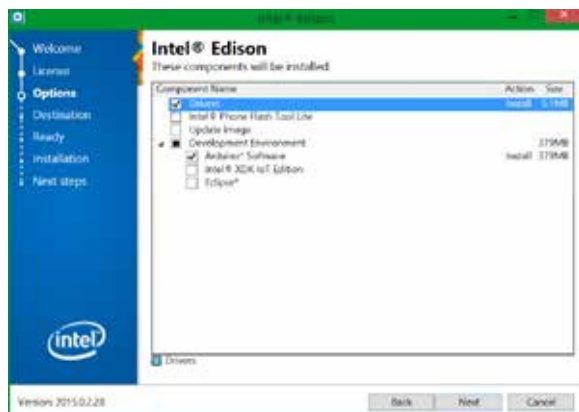
Шаг 4. Если у тебя установлена старая версия JRE и выскочит такое окно, не обращай внимания и жми Next .



Шаг 5. Появится окно с лицензионным соглашением. Выбери I accept the terms of the license и опять нажми Next .



Шаг 6. Теперь выбери компоненты для установки. Поставь галочку возле Drivers, Development Environment и выбери Arduino Software. Возле всех остальных компонентов галочку убери. И опять кликни Next.



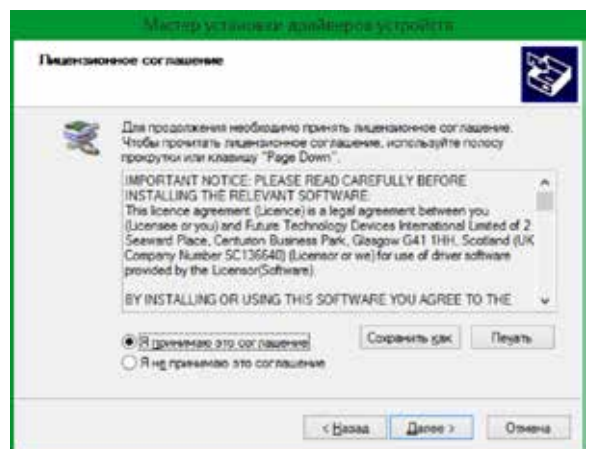
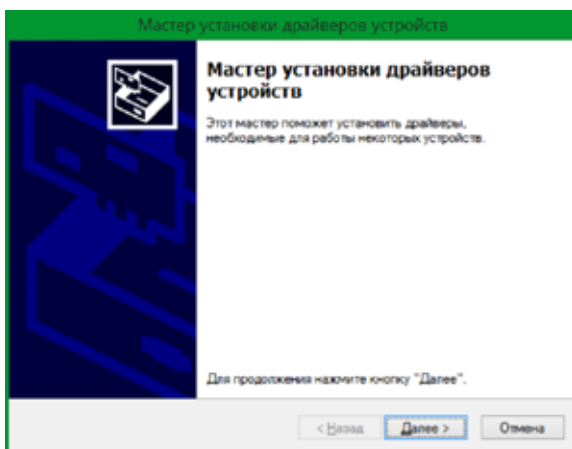
Шаг 7. После этого появится новое окно. Папку установки нельзя изменить, поэтому необходимо иметь как минимум 500 Мб свободного места на диске с . Переходи к следующему шагу, нажав Next .

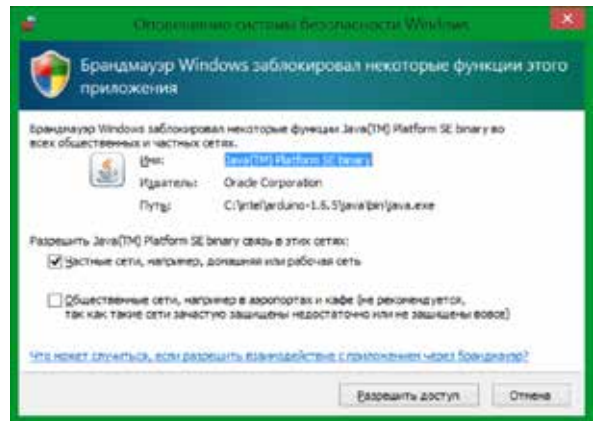
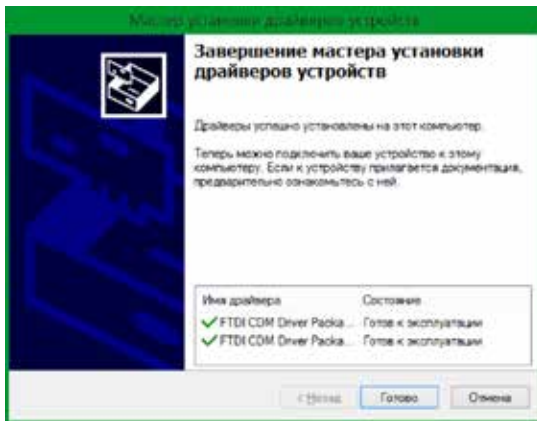


Шаг 8. Появится окно, в котором будут показаны все установленные модули. Если сделал все правильно, нажимай **Next**. Начнется установка.



Шаг 9. В начале и на протяжении установки будут выскакивать окна с установкой дополнительных компонентов («Мастер установки драйверов устройств», «Оповещение системы безопасности Windows» и т. д.). Когда будут появляться такие окна, читай, что в них написано, отвечай везде **Установить**, **Далее**, **Разрешить доступ** и т. д. и продолжай процесс установки.





Шаг 10. После установки нужных программ появится окно об успешном завершении установки. Нажми *Finish*, и программа закроется.

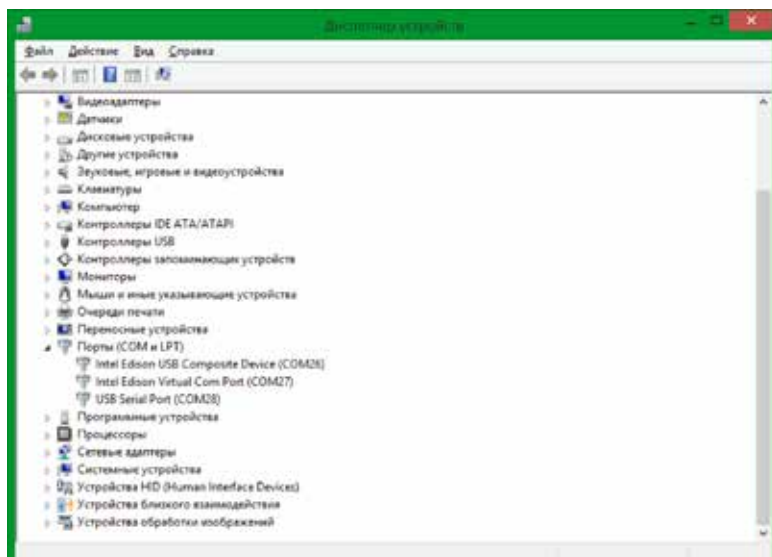


Шаг 11. Теперь тебе надо подключить Intel® Edison и убедиться, что он распознается в системе. Еще раз проверь, чтобы черный микропереключатель был сдвинут в сторону двух микро-USB-портов платы Intel® Edison (Глава 4). Подключи два микро-USB-кабеля к Intel® Edison, а с другой стороны подсоеди́ни их к двум USB-портам компьютера. Запомни: всегда сначала подключа́й центральный микро-USB-кабель, а потом крайний.



После подключения на плате Intel® Edison должен загореться зеленый светодиод. Если у тебя ноутбук, под соедини его к зарядке, чтобы USB-порты компьютера выдавали максимальное напряжение и ток.

Шаг 12. Подожди пару минут, чтобы некоторые драйвера автоматически установились, а потом зайди в «Диспетчер устройств». (В Windows 8–10 нажми сочетание клавиш Win+X и в появившемся списке найди Диспетчер устройств»).



В выпавшем окне найди вкладку «Порты» (COM и LPT). Раскрой выпадающий список. Там должны быть два вида COM-портов со словами Intel Edison в названии (Intel Edison USB Composite Device, Intel Edison Virtual Com Port) и COM-порт USB Serial Port.

Возможна ситуация, при которой порт Intel Edison USB Composite Device не появится. Не обращай на это внимания. Главное, чтобы были два остальных порта. Если это так, переходи к следующим действиям, запомнив, под каким номером указан COM-порт Intel Edison Virtual Com Port. В моем случае это COM24. У тебя, скорее всего, будет другой. Также запомни номер под которым находится USB Serial Port. На моем компьютере это COM12. Данный порт понадобится при работе с платой при помощи программы Putty, через терминал. Это для продвинутого пользователя, которым ты обязательно станешь, но чуть позже.

Если у тебя нет одного из нужных COM-портов (Intel Edison Virtual Com Port и USB Serial Port) или ни один порт не появился, попробуй отсоединить плату Intel® Edison от компьютера и подключить заново, ожидая после повторного подключения 2–3 минуты. Если необходимые COM-порты так и не появились, пиши нам в службу поддержки!

5.2. Вариант № 2 ОС Mac OS X

Пользователям операционной системы OS X нужно выполнить все действия, которые указаны на странице по этой ссылке:

<http://tiny.cc/a1-edison-start-os-x>.

ГЛАВА 6

СРЕДА РАЗРАБОТКИ ПРОГРАММ

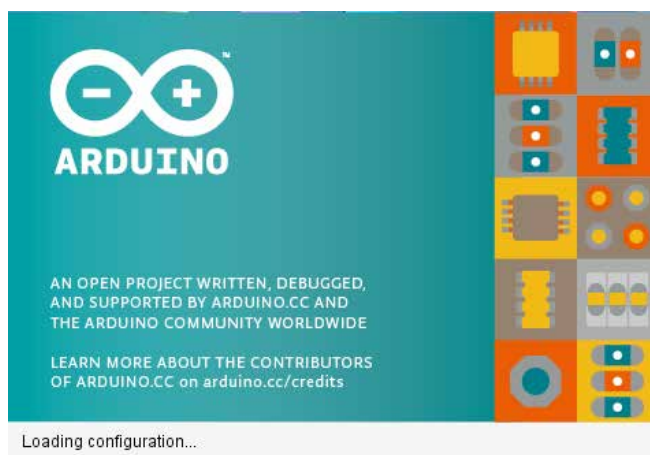
ARDUINO IDE

Любая интегрированная среда разработки необходима для написания программ и их отладки.

Arduino IDE состоит из встроенного текстового редактора программного кода, области сообщений, окна вывода текста (консоли), панели инструментов с кнопками часто используемых команд и меню. Для загрузки программ и связи с платой среда разработки подключается по COM-порту к Intel® Edison.

Arduino IDE – это действенный и простой инструмент для создания программ и загрузки их на плату. Хорошее описание Arduino IDE ты найдешь здесь: <http://tiny.cc/a1-arduino-ide>.

Открой папку с недавно установленными программами и запусти Arduino IDE. Либо зайди в папку установки. Если ты использовал инструкцию для x64, то это папка `C:\Intel\arduino-x.x.x` (крестиками помечены цифры версии приложения Arduino IDE, т. к. они могут быть разные). Запусти `arduino.exe`. Если ты использовал инструкцию x32, то зайди в папку установки программы, которую ты создавал, и запусти `arduino.exe`. Сделай ярлык на рабочем столе этого приложения. Arduino станет основной программой на протяжении всей работы с набором и книгой. После запуска появится окно загрузки приложения.



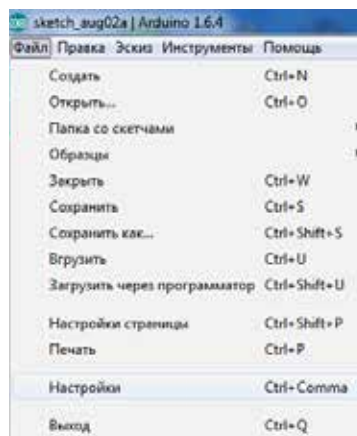
После того как приложение загрузится, появится рабочая область среды программирования Arduino IDE.



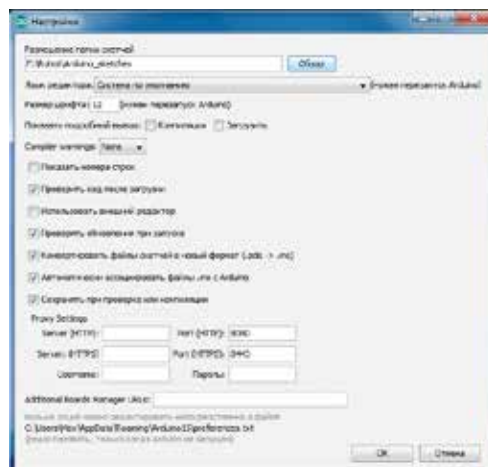
По старой традиции робототехников заведи на своем компьютере папку Robot в корне любого диска, где будут храниться программы и другие материалы по робототехнике. Программы в Arduino IDE принято называть скетчами.

В папке Robot создай папку Arduino_sketches. В ней будут храниться твои программы для Arduino IDE и программы-примеры для набора, которые ты скачаешь чуть позже.

Теперь заверши настройку среды, указав путь для Arduino IDE, где будут храниться программы. Открой первую вкладку «Файл».

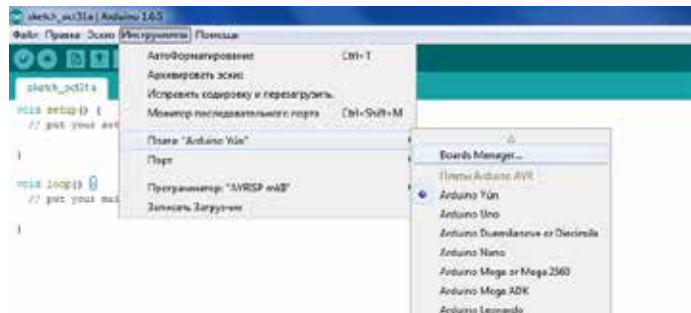


Внизу списка открой вкладку «Настройки». В верхнем окне «Размещение папки скетчей» укажи путь к папке Arduino_sketches. Это можно сделать вручную либо через кнопку Обзор.



Если у тебя Arduino на английском, поменять язык можно во второй строке «Язык редактора». По умолчанию выбирается язык твоей операционной системы, но можно выбрать любой другой. В этом же окне можно поменять размер шрифта (третья строка), если символы на экране кажутся тебе слишком маленькими.

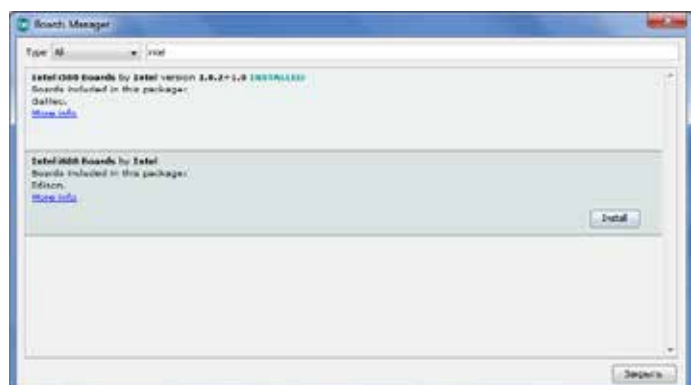
Возможно, в среде Arduino IDE не будет предустановлена плата Intel® Edison. Открой вкладку «Инструменты-Плата» и посмотри, есть ли там Intel® Edison.



Если нет, то открой Boards Manager в той же вкладке «Плата» (находится в самом верху). Появится такое окно.



Введи в поиске слово intel. Появятся два раздела. Кликни по Intel i686 Boards (нижний раздел) и нажми на кнопку Install. Начнется скачивание и установка файлов для платы Intel® Edison.



Переходи к следующей главе. Дополнительная информация и возможные правки к главе доступны по ссылке: <http://tiny.cc/a1-edit-c6>.

ГЛАВА 7

КРАТКАЯ ТЕОРИЯ ПО ПРОГРАММИРОВАНИЮ В СРЕДЕ ARDUINO IDE

Теперь вкратце рассмотрим теорию по программированию в среде Arduino IDE. Язык программирования Arduino, на котором идет создание программ, основан на языке Wiring. Во многом он похож на язык C++, но с небольшими изменениями и дополнениями.

Любознательные могут проследовать по этой ссылке, чтобы получить больше информации <http://tiny.cc/a1-arduino-language>.

Дополнительная информация и возможные правки к главе доступны по ссылке: <http://tiny.cc/a1-edit-c7>.

7.1. Структура программы

Все элементы программы в Arduino IDE крайне подробно описаны по этой ссылке: <http://tiny.cc/a1-arduino-programming>. Ниже я приведу краткую памятку по основным элементам.

Каждый Arduino-скетч имеет две необходимые функции: `void setup()` и `void loop()` (также называемые подпрограммами).

```
void setup()  
{  
  //...  
}
```

Весь код, который находится внутри фигурных скобок данной функции, будет выполняться только один раз. Используется для инициализации переменных, определения режимов работы выводов, запуска используемых библиотек и т. д.

```
void loop()  
{  
  //...  
}
```

Эта функция выполняется после того, как завершится выполнение кода в функции `setup()`. В функции `void loop()` пишется основной код программы. Используйте эту функцию, чтобы активно управлять своей платой. Когда функция `void loop()` выпол-

нится один раз, она будет выполняться снова и снова, до тех пор, пока плата не будет перезагружена либо не будет отключено питание от Intel Edison. Другими словами, это бесконечный цикл.

7.2. Синтаксис

Одним из требований к языку программирования является требование к форматированию кода. Если ты помнишь следующее, то писать программы будешь с легкостью.

// (однострочный комментарий)

Очень полезно оставлять пометки для себя, чтобы разобраться, что значит каждая строка твоего кода, когда ты долго с ним не работал и подзабыл суть программы. Чтобы оставить однострочный комментарий, напечатай два символа «/» друг за другом, и все, что будет напечатано после них до конца строки, будет игнорироваться программой и не выполняться.

/* */ (многострочный комментарий)

Если ты решил, что тебе надо многое рассказать про свою программу и весь текст не умещается в одной строке, ты можешь использовать данные символы многострочного комментария. Все, что будет написано между этими двумя символами, также будет игнорироваться программой.

{ } (фигурные скобки)

Используются, чтобы определить, где блок кода начинается «{» и заканчивается «}». Используются в функциях, операторах и циклах.

; (точка с запятой)

Каждая строка кода должна оканчиваться точкой с запятой (пропуск этого символа в коде нередко становится причиной, по которой программа не хочет компилироваться и загружаться на плату).

7.3. Переменные

Программа – это набор инструкций для компьютера по перемещению набора цифр в нужной пользователю последовательности. Переменные служат для того, чтобы совершать данные перемещения.

int (integer)

Такая переменная является основной «рабочей лошадкой» программы и хранит число в двух байтах (16 бит), т. е. занимает места в программе ровно 2 байта. Хранимое число должно быть целым (без дробной части), а величина его может составлять от – 32 768 до 32 767.

long (long)

Такая переменная используется тогда, когда размера переменной `int` не хватает для хранения числа. То есть число 32,769 уже не может храниться в переменной `int`, поэтому для него необходима переменная `long`, занимающая в RAM-памяти 4 байта (32 бита), и диапазон чисел, которые могут храниться в переменных данного типа, меняется от $-2\,147\,483\,648$ до $2\,147\,483\,647$.

boolean (boolean)

Эта переменная отличается от своих предшественниц. Она может принимать только логические типы данных – не числа, а состояния `true` или `false`. Занимает в памяти 1 байт.

Пример:

```
boolean x=false;
```

Запись означает: переменной с именем `x` присвоить значение `false`. Очень полезна в тех случаях, когда необходимо экономить память контроллера, т. к. занимает в ней всего один байт!

float (float)

Если есть необходимость хранить дробные числа с плавающей запятой, то `float` способна это делать. Занимает 4 байта (32 бита) в памяти и имеет диапазон принимаемых значений от $-3,4028235E+38$ до $3,4028235E+38$. Буква `E` обозначает просто множитель. Поэтому запишем в более понятной форме: от до.

char (character)

Такая переменная хранит один символ, используя ASCII-код. Занимает в памяти 1 байт. А что, если необходимо ввести строку? Строка – это не что иное, как массив символов.

Пример:

```
char myChar='M';
char myChar=77; //Оба варианта одинаковы
```

Поясню тебе про ASCII-код. Все символы на клавиатуре, передаются компьютеру или микроконтроллеру в виде ASCII-кода. То есть нажав на клавишу с английской «М», мы передаем компьютеру число 77. Это означает, что символ `M` закодирован под номером 77 в десятичной системе счисления. Вся таблица ASCII для ознакомления представлена здесь: <http://tiny.cc/a1-ascii>.

7.4. Арифметические операторы

Операторы в программировании используются, чтобы работать с переменными. Операторы работают в программе, как и в обычной математике.

= (оператор присваивания) Присваивает переменной слева от оператора значение переменной или выражения, находящихся справа. Например, `x = 20;` // переменной `x` присвоено значение 20.

% (оператор остаток от деления) Возвращает остаток от деления одного целого числа на другое целое число. Например, `x = 7 % 5;` // `x` имеет значение 2. Число 5 может один раз поместиться в числе 7, поэтому остаток 2.

+ (оператор сложения)

- (оператор вычитания)

***** (оператор умножения)

/ (оператор деления)

7.5. Операторы сравнения

Такие операторы используются для логического сравнения переменных.

== (равно) Например, `7==10` неверно, `7==7` верно.

!= (не равно) Например, `7!=10` верно, `7!=7` неверно.

< (меньше чем) Например, `10<7` неверно или `7<7` неверно, а вот `9<10` верно.

> (больше чем) Например, `13>15` неверно, `13>13` неверно, `13>7` верно.

>= (больше либо равно) Например, `15>=13` – это верное выражение. Хотя 15 не равно 13, но 15 больше 13. А вот `12>=13` – это неверное выражение.

<= (меньше либо равно) Например, `12<=12` – это верное выражение, т.к. выполняется одно из двух условий. 12 равно 12, но не выполняется условие, что 12 меньше 12.

7.6. Управляющие операторы

Управляющие операторы предназначены для управления ходом выполнения программы.

Оператор `if` (если)

Запись:

```
if (Условие) {  
    //код программы, если выполняется Условие  
}
```

Такой оператор используется, чтобы выполнить код, который написан в фигурных скобках после оператора, когда условие в круглых скобках истинно.

Пример:

```
int x=5;  
int y=4;  
boolean check=false;  
if (x>y) { check=true; }
```

Условие выполняется, т. к. $5 > 4$, следовательно, и код в фигурных скобках тоже выполняется, поэтому в переменную `check` записывается `true`.

Оператор `if(если) ... else(тогда)`

Запись:

```
if (Условие) {
    //код программы, если выполняется Условие
}
else {
    //код программы, если не выполняется Усло-
    вие
}
```

Пример:

```
int x=5;
int y=4;
boolean check=false;
if (y>x) {
    check=true;
}
else {
    check=false;
}
```

Первое условие не выполняется, т. к. $4 < 5$, следовательно, код в фигурных скобках после `if` тоже не выполняется. Переходим к `else` и выполняем условие по записи в переменную `check` значения `false`.

Оператор `if ... else if ... else ...`

Запись:

```
if (Условие 1) {
    //код программы, если выполняется Условие 1
}
else if (Условие 2) {
    //код программы, если выполняется Условие 2
}
else {
    //код программы, если не выполняется Условие
    2
}
```

Текущая структура редко используется на начальных этапах программирования. Поэтому для понимания кода и упрощения его вида, мы будем использовать предыдущие две конструкции.

Цикл `for (...)`

Запись:

```
for (initialization; condition; increment) {
    //Код программы }
```

Цикл `for` в основном используется для того, чтобы повторять действие в скобках определенное количество раз до тех пор, пока условие в цикле (`condition`) истинное.

Инициализация (initialization) выполняется самой первой и один раз. Каждый раз в цикле проверяется условие (condition); если оно верно, выполняется код программы в фигурных скобках и приращение (increment), затем условие проверяется вновь. Когда логическое значение условия становится ложным, цикл завершается.

Пример :

```
int m=0;
for (int i=0; i<5; i++) {
  m=m+1;
}
```

По завершении цикла for в переменную m будет записано число 5. Именно столько раз выполнится цикл, пока условие $i < 5$ истинно. А приращение после каждого цикла для i составляло 1, т. к. операция i++ равносильна операции $i = i + 1$. Пройдемся по тому, как это произошло:

1. При первом цикле наш счетчик – это переменная i, она имела значение 0. С этим значением и выполнялась операция в фигурных скобках первый раз. Теперь в m записано значение 1.
2. При втором цикле i равно 1, m равно 2.
3. При третьем цикле i равно 2, m равно 3.
4. При четвертом цикле i равно 3, m равно 4.
5. При пятом цикле i равно 4, m равно 5.
6. Выход из цикла. i равно 5, $5 < 5$ ложь.

Цикл while (до тех пор пока)**Запись :**

```
while (выражение) {
  // Код программы
}
```

while будет вычислять код в цикле непрерывно до тех пор, пока выражение в круглых скобках () не станет ложным. Что-то должно изменить значение проверяемой переменной, иначе выход из цикла while никогда не будет достигнут. Это изменение может происходить как в программном коде (например, при увеличении переменной), так и во внешних условиях (допустим, при тестировании датчика).

Пример :

```
int var=0; //Переменная должна быть объявлена глобально
void setup() {
}
void loop() {
  while(var<100) {
    //Выполняем код программы, повторив его 100 раз
  }
}
```



```
//Каждый раз мы прибавляем к переменной var
единицу
//Цикл while закончится тогда, когда пере-
менная var будет равна 100. //Тогда условие
var<100 станет ложным.
var++;
}
}
```

7.7. Функции

Все функции бывают двух типов. Функции, которые возвращают значение, имеют тип `int`, а которые не возвращают значение – `void`. Я расскажу про стандартные функции в Arduino IDE. Что-бы создавать свои функции, изучи информацию по этой ссылке: <http://tiny.cc/a1-functions>.

Чтобы лучше понимать, как расположены цифровые и аналоговые порты ввода-вывода, рекомендую при прочтении нижеследующего текста смотреть на схему выводов и основных разъемов платы Intel® Edison ([Глава 8](#)), а также на реальную плату Intel® Edison.

Функции по работе с цифровыми сигналами

Прежде, чем переходить к изучению этих функций я скажу пару слов о цифровых сигналах. Цифровой сигнал — это сигнал, который можно представить в виде последовательности дискретных (цифровых) значений. В наше время наиболее распространены двоичные цифровые сигналы в связи с простотой кодирования и использованием в двоичной электронике. На рисунке ниже представлен типичный цифровой сигнал, в котором есть два состояния: HIGH, или значение логической единицы, и LOW, или значение логического нуля. Подробнее про цифровой сигнал написано здесь: <http://tiny.cc/a1-digital>.

Функция `pinMode (pin, Mode)`



Эта функция устанавливает режим работы заданного пина (далее может быть именован как вывод) как входа или как выхода. Mode может быть либо `OUTPUT` либо `INPUT`. Не возвращает значение. Желательно использовать эту функцию в структуре `void setup()`.

Пример:

```
pinMode (3, OUTPUT) ;
```

Такая запись говорит, что цифровой вывод 3 (D3) настроен на выход.

Функция `digitalWrite(pin, value)`

Когда вывод настроен на выход (OUTPUT) функцией `pinMode()`, он может выдавать значение HIGH, т. е. на нем будет выставлено значение высокого напряжения 5 В либо LOW, в таком случае напряжения на нем не будет. Не возвращает значение.

Пример:

```
digitalWrite(13, HIGH);
```

Это говорит о том, что на цифровой вывод 13 подано напряжение 5 В, он принял значение HIGH.

Пример:

```
digitalWrite(13, LOW);
```

А это значит, что на цифровом выводе под номером 13 напряжения нет, он принял значение LOW.

Функция `digitalRead(pin)`

Когда пин настроен как вход (INPUT), ты можешь использовать данную функцию, чтобы узнать, в каком состоянии находится вывод. Если напряжение на нем 5 В, будет возвращено значение HIGH, если 0В, то LOW. Возвращает значение HIGH или LOW в зависимости от состояния вывода, к которому она была применена. В примере будет просчитано состояние цифрового вывода под номером 6.

Пример:

```
int inPin=6; // Переменная inPin содержит  
номер вывода на плате  
int val = digitalRead(inPin); // считываем  
значение с вывода inPin
```

Функции по работе с аналоговыми сигналами

Intel® Edison является цифровым устройством, однако у него есть возможность оперировать и с аналоговыми сигналами.

Я коротко опишу как взаимодействовать с сигналами, если они являются не цифровыми, а аналоговыми. На рисунке ниже представлен аналоговый сигнал.

У него, как ты видишь, отсутствует состояние HIGH или LOW, если сравнивать его с цифровым сигналом. Однако аналоговый сигнал имеет значение минимума и максимума, а все величины между двумя этими пределами меняются по определенной функции, которая имеет бесконечно много значений в указанных ранее пределах. Таким образом, аналоговый сигнал никак не может быть обработан цифровой электроникой, которая оперирует лишь двумя состояниями: 0 и 1. Подробнее про аналоговый сигнал прочти здесь: <http://tiny.cc/a1-analog>.



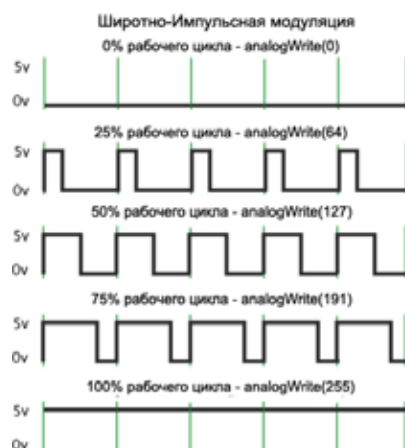
Для обеспечения совместимости аналогового сигнала с цифровой электроникой используется АЦП. АЦП – это аналого-цифровой преобразователь, который трансформирует аналоговый сигнал в цифровой. АЦП выглядит как небольшая микросхема, которая устанавливается в исходную схему устройства. На вход АЦП поступает аналоговый сигнал, а на выходе получается цифровой, понятный цифровой электронике. Таким же образом можно преобразовать цифровой сигнал в аналоговый. Для этого существует ЦАП, или цифро-аналоговый преобразователь. Подробнее про АЦП и ЦАП ты можешь почитать по ссылкам: <http://tiny.cc/a1-adc> и <http://tiny.cc/a1-dac>.

Широтно-импульсная модуляция

Перед тем как ознакомиться с функцией `analogWrite()`, тебе необходимо узнать, что такое ШИМ (широтно-импульсная модуляция), или PWM (pulse-width modulation).

Широтно-импульсная модуляция – это операция получения изменяющегося аналогового значения посредством цифровых устройств. На выводе (пине) устройства генерируется последовательность прямоугольных импульсов определенной длительности. Такие импульсы – это сигналы, которые постоянно переключаются между максимальным и минимальным значениями. Широтно-импульсная модуляция моделирует напряжение между максимальным значением (5 В) и минимальным (0 В), изменяя при этом только длительность времени включения 5 В относительно включения 0 В. Длительность включения максимального значения называется шириной импульса. Для получения различных аналоговых величин изменяется ширина импульса. При достаточно быстрой смене периодов включения-выключения можно подавать постоянный сигнал между 0 и 5 В на светодиод, управляя яркостью его свечения.

На графике зеленые линии отмечают постоянные временные периоды. Длительность периода обратно пропорциональна частоте ШИМ. Т. е. если частота ШИМ составляет 500 Гц, то зеленые линии будут отмечать интервалы длительностью в 2 миллисекунды каждый. Вызов функции `analogWrite()` с масштабом 0–255 означает, что значение `analogWrite(255)` будет соответствовать 100%-ному рабочему циклу (постоянное включение 5 В), а значение `analogWrite(127)` – 50%-ному рабочему циклу.



Увеличение или уменьшение ширины импульса на светодиоде будет увеличивать или уменьшать его яркость. ШИМ – это очень важная деталь в робототехнике и не только. Благодаря ШИМ можно управлять скоростью вращения электродвигателя или углом поворота сервопривода, а также множеством других электро-механических устройств, способных изменять свое состояние. Теперь ты знаешь, что такое ШИМ. Тебя смело можно допускать к знакомству с дальнейшим материалом книги. Правда, будет лучше, если ты согласишься посмотреть видео про ШИМ, по этой ссылке: <http://tiny.cc/a1-pwm>.

Функция `analogWrite (pin, value)`

Генерирует аналоговую величину (ШИМ-волну) на порт входа/выхода. Функция может быть полезна для управления яркостью подключенного светодиода или скоростью электродвигателя. После вызова `analogWrite()` на выходе будет генерироваться постоянная прямоугольная волна с заданной шириной импульса до следующего вызова `analogWrite` (или вызова `digitalWrite` либо `digitalRead` на том же порту входа/выхода). Частота ШИМ-сигнала приблизительно 490 Гц.

Переменная `pin` обозначает порт входа/выхода, на который подается ШИМ-сигнал.

А переменная `value` обозначает период рабочего цикла, значение которого находится между 0 (полностью выключено) и 255 (сигнал подается постоянно).

Пример :

```
analogWrite(5, 0);
```

Это означает, что ШИМ-сигнал на цифровом выводе 5 не генерируется. Выставлено значение 0. На рисунке выше это первый график.

Пример :

```
analogWrite(5, 120);
```

Генерируется ШИМ-сигнал на цифровом выводе 5 с периодом рабочего цикла (скважностью) приблизительно ~50%. На рисунке выше это третий график.

Пример :

```
analogWrite(5, 255);
```

Генерируется ШИМ-сигнал на цифровом выводе 5 с периодом рабочего цикла (скважностью) приблизительно 100%. Пятый график.

Функция `analogRead (pin)`

Когда аналоговый вывод (пин) настроен на вход, ты можешь считать значение напряжения относительно опорного напряжения AREF-платы. Возвращает значение от 0 до 1024. Если функция возвращает 0, то напряжение, подаваемое на данный аналоговый пин, соответствует 0 В, если выдает 1024, это соответствует 5 В. Аналоговый сигнал в виде напряжения подается на АЦП, преобразуется в цифровой, а затем считывается микроконтроллером.

Пример:

```
int val = analogRead(0);
```

Считываем значение напряжения с аналогового пина 0 в переменную val.

Функции по работе со временем

Эти функции не смогут остановить течение времени во Вселенной, но остановить выполнение программы на некоторое время или измерить время в программе они могут.

Функция millis()

Эта функция возвращает количество миллисекунд с момента начала выполнения текущей программы на плате. Функция очень полезна, когда необходимо измерить время выполнения всей программы или отдельной ее части. В примере ниже будет измерено время выполнения программы.

Пример:

```
void setup() {
  // Инициализируем последовательный порт на
  // скорости 9600 бод.
  Serial.begin(9600);
}

void loop() {
  int a=millis(); // Получаем первое значение
  // времени
  // Здесь может выполняться некоторый код
  // В нашем случае будет считано состояние
  // вывода
  int val = digitalRead(4); // Выполняется
  // функция digitalWrite()

  int b=millis(); // Получаем второе значение
  // времени
  int c=b-a; // В переменную c записывается
  // разница между двумя значениями функции
  // millis() b и a
  // В монитор последовательного порта выво-
  // дится время
  // за которое была выполнена функция
  // digitalWrite()
  Serial.println(c);
}
```

Функция delay()

Функция останавливает выполнение программы на заданное количество миллисекунд.

Пример:

```
void setup() {
}
```

```
void loop() {
  int val = digitalRead(4); // Выполняется
  функция digitalWrite()

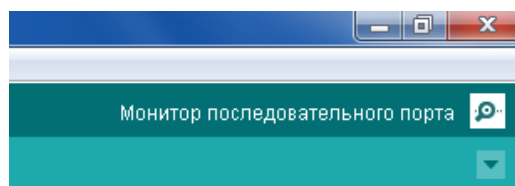
  /* delay(1000) создает задержку в 1 секун-
  ду либо 1000 мс. Функция принимает значения
  только в миллисекундах */
  delay(1000);
}
```

7.8. Отладка программ в Serial

Понять, какое действие выполняет в данный момент программа, тебе поможет монитор последовательного порта. Это своего рода дисплей твоего устройства, только в данном случае его роль играет монитор твоего компьютера.

Набор функций **Serial** служит для связи устройства Intel® Edison с компьютером или другими устройствами, поддерживающими последовательный интерфейс обмена данными. Для обмена данными **Serial** используют цифровые порты ввода/вывода 0 (RX) и 1 (TX), а также USB-порт. Важно учитывать, что если ты используешь функции **Serial**, то нельзя одновременно с этим использовать порты 0 и 1 для других целей.

Чтобы открыть монитор, наведи стрелку мыши на кнопку с лупой в верхнем правом углу среды IDE Arduino либо нажми сочетание клавиш Ctrl+Shift+M. Важно, чтобы скорость последовательного порта, прописанного в программе, совпадала со скоростью, которая выставлена в окне монитора последовательного порта в среде Arduino (нижний правый угол окна монитора последовательного порта). В противном случае, на экран будут выводиться непонятные символы.



Пример 1:

Простой вывод текста на монитор последовательного порта

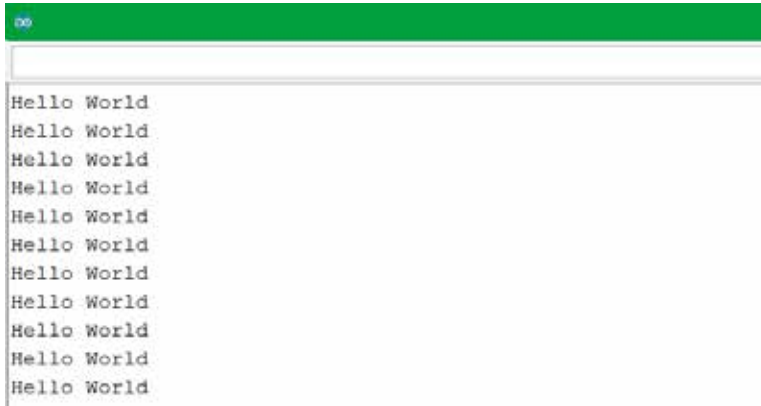
Название программы A1_Serial_1.ino

```
void setup()
{
  // Инициализируем последовательный порт на
  скорости 9600 бод.
  Serial.begin(9600);
}
```

```

void loop()
{
  /* Выводит на экран строку с текстом Hello
  World и переходит на новую строку */
  Serial.println("Hello World");
  /* delay(1000) создает задержку в 1 секун-
  ду либо 1000 мс. Функция принимает значения
  только в миллисекундах */
  delay(1000);
}

```



```

Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World

```

Замечу, что любой текст, для вывода на монитор необходимо брать в двойные кавычки.

Пример 2:

Вывод значения переменной на монитор последовательного порта

Название программы A1_Serial_2.ino

```

void setup()
{
  // Инициализируем последовательный порт на
  скорости 9600 бод.
  Serial.begin(9600);
}

void loop()
{
  /* Создаем целочисленную переменную под на-
  званием переменная типа int */
  int переменная = 19;

  /* Выводим на экран значение переменной и
  переходим на новую строку */
  Serial.println(переменная);

  /* delay(1000) создает задержку в 1 секун-
  ду либо 1000 мс. Функция принимает значения
  только в миллисекундах */
  delay(1000);
}

```

Важные особенности Intel® Edison при использовании Serial

На самом деле эта плата имеет следующие Serial-порты:

1. Serial

- Возможные названия порта: Multi-gadget, Firmware Programming, Serial console, OTG-порт.
- Служит для загрузки скетчей Arduino в плату, доступа к внутреннему хранилищу платы.
- Имя в Arduino IDE: Serial.

2. Serial1

- Возможные названия порта: UART1, TTL-level-порт.
- Портом используются цифровые пины платы 0 и 1.
- Имя в Arduino IDE: Serial1.

3. Serial2

4. SerialGPS

Подробнее про Serial в Intel® Edison читай здесь <http://tiny.cc/a1-edison-serial>.

Если необходимо управлять по последовательному порту устройствами, которые управляются через пины платы 0(RX) и 1(TX), следует использовать Serial1, а не Serial. Например, тебе нужно отправить команду на устройство (драйвер управления сервоприводами) в текстовом виде. Устройство подключено к плате через пины 0, 1 и GND.

Пример :

```
void setup()
{
  // Инициализируем последовательный порт
  Serial1 на скорости 9600 бод.
  Serial1.begin(9600);
}

void loop()
{
  // Отправляем команду на устройство
  Serial1.print("#1 P1500 <cr>");

  /* delay(1000) создает задержку в 1 секунду
  либо 1000 мс. Функция принимает значения
  только в миллисекундах */
  delay(1000);
}
```

Подробнее про все структуры, переменные и операторы ты можешь прочитать, пройдя по этой ссылке: <http://tiny.cc/a1-arduino-reference>.

ГЛАВА 8

ПОЛЕЗНАЯ ИНФОРМАЦИЯ ПЕРЕД НАЧАЛОМ РАБОТЫ

8.1. Памятка по работе с набором и платой

Дополнительная информация и возможные правки к главе доступны по ссылке: <http://tiny.cc/a1-edit-c8>.

Не подавай на аналоговые и цифровые выводы основной платы напряжение больше 5 В. Для подачи напряжения на выводы платы, соответствующие выводы должны быть сконфигурированы как INPUT (конфигурация необходима только для цифровых выводов).

Не подсоединяй вывод цифрового пина к земле (вывод GND) без нагрузки (нагрузкой может выступать электромотор, светодиод, резистор и т. д.), когда он сконфигурирован как выход и на этот цифровой вывод подведено напряжение, т. е. записано значение HIGH. Так же и с аналоговым выводом, который сконфигурирован как выход.

Не подсоединяй выводы 5 В и 3.3 В к выводам GND без нагрузки! Это приведет к короткому замыканию, которое может вывести плату Intel® Edison из строя!

В любых соединениях сначала подключай землю (GND-вывод), а потом замыкай схему, подсоединяя плюс (5 В, 3,3 В либо аналоговый или цифровой вывод). Во

всех схемах, все выводы GND с любых устройств должны быть объединены!

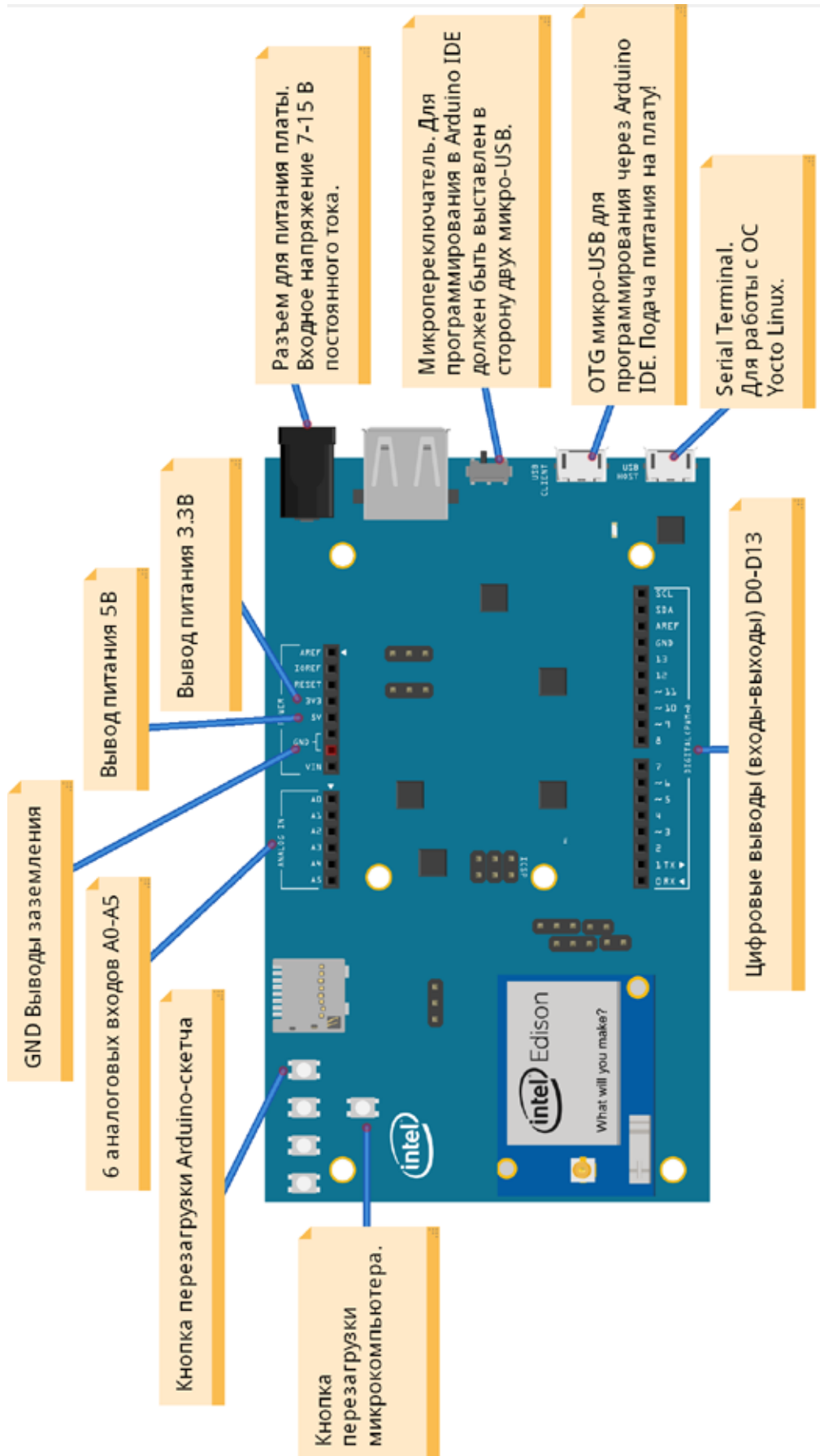
Настройку любых выводов желательно производить в структуре `void setup() { }`.

Цифровые выводы (пины) 0–1 (D0–D1) не рекомендуется использовать в своих схемах, т. к. они задействованы для передачи данных по серийному порту на компьютер. С их помощью происходит передача информации в монитор последовательного порта Serial. Если ты их задействуешь, то не сможешь получать информацию от платы на свой компьютер. Это не критично для тех задач, которые не требуют использования Serial.

В Intel® Edison широтно-импульсную модуляцию по умолчанию поддерживают цифровые выводы с номерами 3, 5, 6, 9, и максимальное количество поддерживаемых цифровых ШИМ-выводов равно 4. При помощи перемычек возможно поменять номера цифровых ШИМ-выводов на другие цифровые выводы под номерами 3, 5, 6, 9, 10, 11. Подробнее о том, как это сделать написано здесь: <http://tiny.cc/a1-edison-pwm>.

Аналоговые входы (analog pins) также могут быть использованы как цифровые входы/ выходы (digital pins). Обращение к ним идет по номерам от 14 (для аналогового вывода 0) до 19 (для аналогового вывода 5).

8.2. Схема выводов и основных разъемов платы Intel® Edison



2 ПРАКТИКА

ГЛАВА 9

СОЗДАНИЕ СХЕМ И ИХ ПРОГРАММИРОВАНИЕ

Схемы уровня сложности А

В этой части Книги новичка ты перейдешь к самому главному, а именно, будешь на практике применять теоретические знания, которые получил из предыдущего раздела. Ты научишься программировать микрокомпьютер Intel® Edison!

В ходе работы с [Главами 9, 10, 11 и 12](#) у тебя могут возникать различные ошибки. Способы устранения многих из них описаны в [Главе 13](#). Дополнительная информация и возможные правки к главе доступны по ссылке: <http://tiny.cc/a1-edit-c9>.

Все изображения схем и исходные файлы ты можешь скачать по ссылке: <http://tiny.cc/a1-pics>.

Для начала работы тебе надо скачать скетч-примеры с исходным кодом для всех схем уровня А, А+ и В.

Ссылка для скачивания:

<http://tiny.cc/a1-arduino-code> либо

https://github.com/mrobotby/A1_Arduino_sketches.

После того как ты будешь переадресован с этой ссылки на сайт `github.com`, в нижнем правом углу, кликни на кнопку `Download ZIP`. Начнется скачивание архива. После скачивания заходи в загрузки. Разархивируй архив без изменения названия папки. Т. е. после разархивирования должна получиться папка `A1_Arduino_sketches-master`.

Теперь убедись, что у тебя закрыта `Arduino IDE`. Вырежи или скопируй папку `A1_Arduino_sketches-master` в папку `Arduino_sketches`, которую ты создал в начале [Главы 6](#). Все программы имеют комментарии к коду, чтобы ты мог без труда разобраться, как работает та или иная программа. Внимательно изучай пояснения к коду, а не бездумно загружай скетч на плату. Доработка скетчей приветствуется!

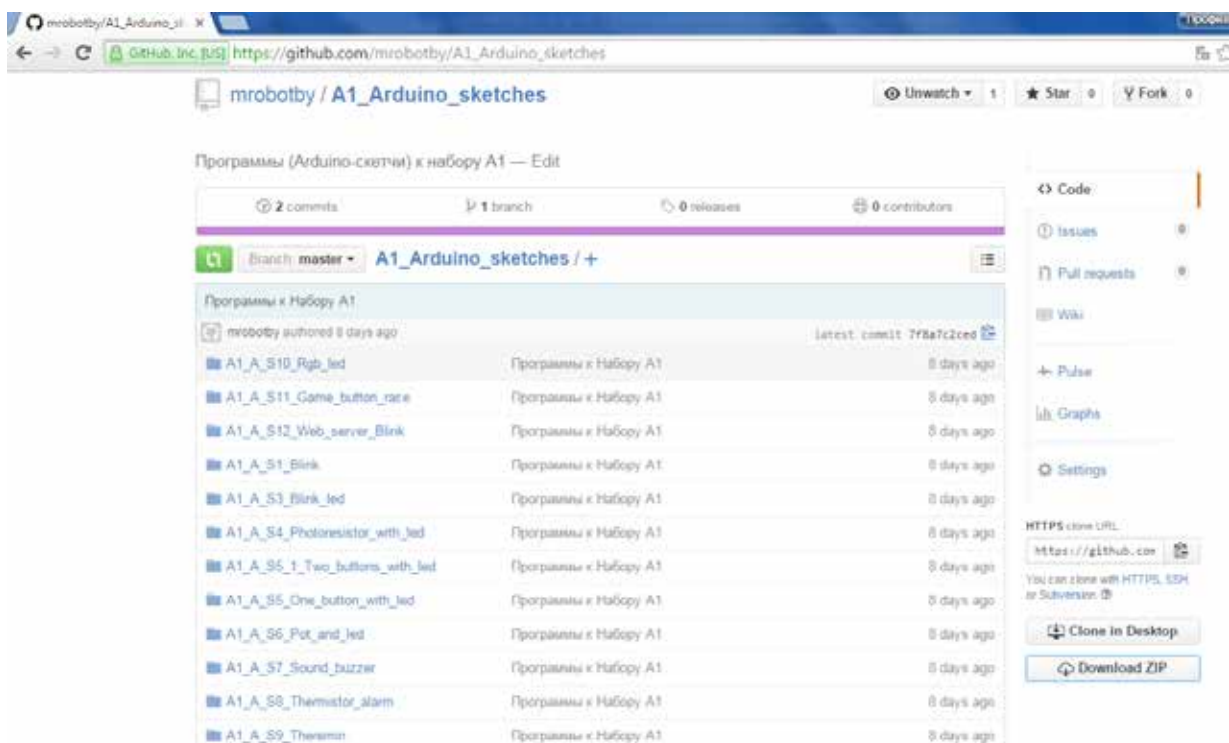


СХЕМА 1 МОРГАНИЕ ВСТРОЕННЫМ СВЕТОДИОДОМ

Итак, перейдем к самой первой схеме, которую проходит любой новичок, начинающий работать с платами разработки: морганию светодиодом с помощью программы. Моргание светодиодом, или Blink, – это как бессмертный Hello World для начинающих программистов. На плате Arduino-расширения Intel® Edison есть встроенный светодиод, который подключен к 13-му цифровому выводу (D13). Будем считать встроенный светодиод на плате небольшим помощником в отладке программ.

Работая над Схемой 1, мы изучим основы загрузки и программирования для Intel® Edison и будем моргать встроенным светодиодом на плате с помощью программы, загруженной в Intel® Edison.

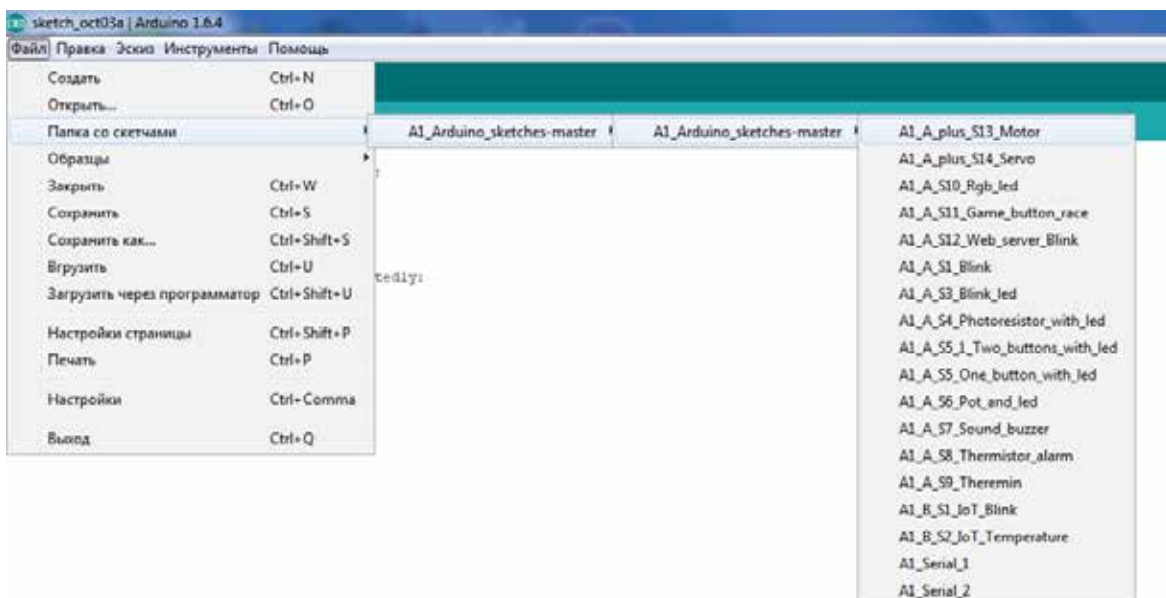
Что необходимо Кабель микро-USB.

Плата Intel® Edison. Во всех остальных схемах она будет необходима по умолчанию, и далее я об этом писать не буду.

Действие 1. Подключи микро-USB-кабель к OTG-порту платы Intel® Edison (микро-USB посередине платы, смотри [Главу 8](#), пункт 8.2) и подключи кабель к компьютеру.

Действие 2. Открой папку со всеми проектами.

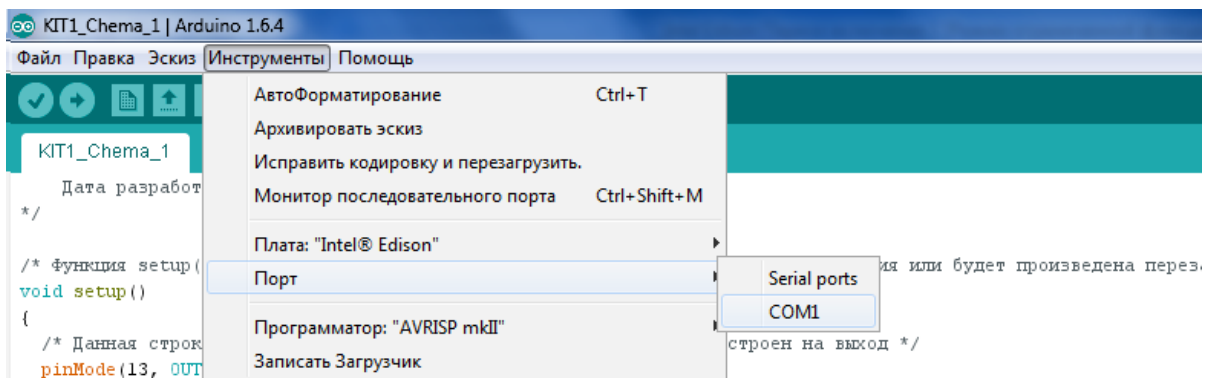
Открой файл с названием A1_A_S1_Blink.ino в одноименной папке. После чего Arduino IDE его распознает и откроет. Если же файл не открывается, открой Arduino IDE самостоятельно: вкладка «Файл», потом «Открыть», далее указываем путь к нашей программе. Прочти все комментарии, которые приведены в тексте программы, чтобы лучше понять, как она работает.



Кстати, если выбрать «Файл» (File), а чуть ниже навести курсор на «Папка со скетчами» (Sketchbook, если английская версия), то откроется твоя папка с программами Arduino_sketches. Наведя на нее курсор, можно раскрыть еще один выпадающий список с программами и найти нужную тебе. Запомни данную операцию, т. к. она очень удобна, смотри картинку выше.

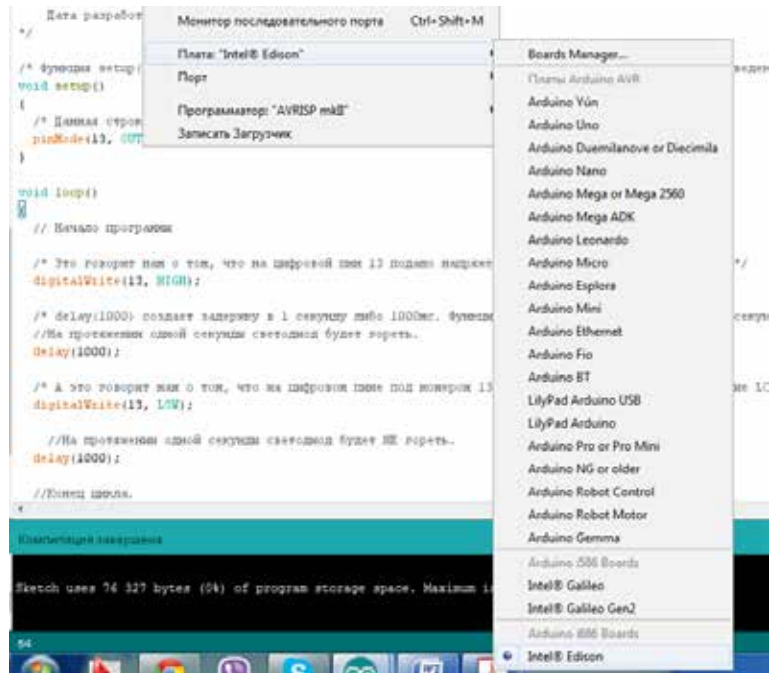
Первое время я очень рекомендую всем набирать код вручную, хоть это и кажется нудным и бесполезным занятием. Когда ты набираешь программу сам, а не копируешь ее, ты подсознательно запоминаешь ее структуру и конструкции. Как результат, ты более осознанно начнешь программировать в дальнейшем. Я снабдил каждый скетч комментариями – может, ты уже все запомнил, но нелишним будет написать, что значит каждая строка кода в программе.

Действие 3. Вспоминай, под каким номером был указан COM-порт, подписанный как Intel Edison Virtual Com Port. Если забыл, то возвращайся к Шагу 14 (Инструкция № 1) или Шагу 12 (Инструкция № 2).



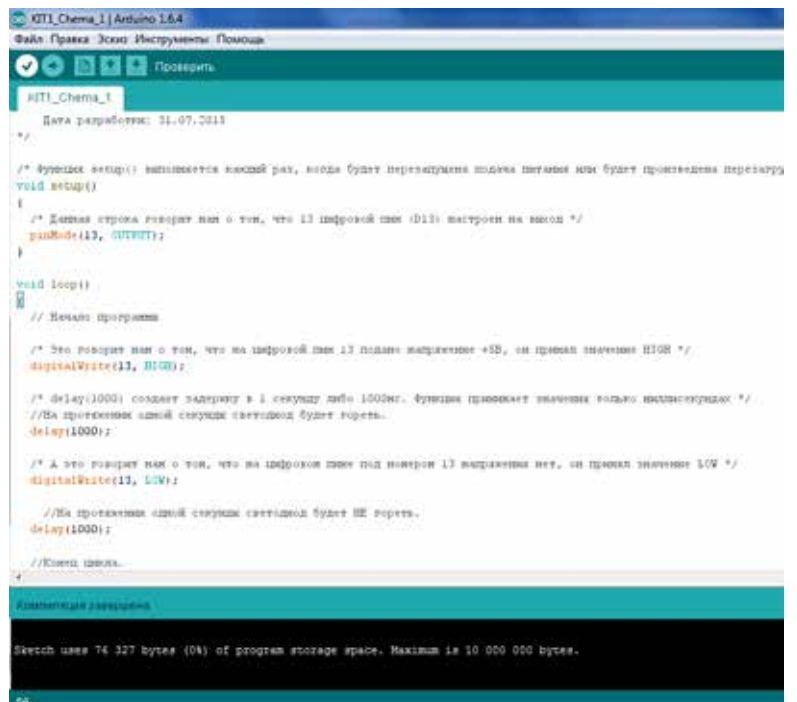
Действие 4. Выбери нужный номер порта – ты определил его, выполняя Действие 3. Может возникнуть такая ситуация, что нужный COM-порт не появился, подожди пару минут. Ничего не происходит? Тогда отсоедини плату и подсоедини обратно. Подожди пару минут. Если по-прежнему ничего, отсоедини плату и перезагрузи компьютер. После перезагрузки подсоедини плату, запусти среду Arduino IDE. Нужный порт должен появиться. Так и не появился? Пиши нам в службу поддержки.

Действие 5. Теперь необходимо выбрать плату Intel® Edison, чтобы среда Arduino IDE знала, что мы собираемся программировать плату Intel® Edison, а не какую-то другую. Выбери Intel® Edison. Она находится в самом низу списка.



Действие 6.

Далее нужно проверить программу на ошибки, иначе она не загрузится на плату, а после этого скомпилировать в понятный для микроконтроллера вид. Для этого надо нажать на круглую зеленую кнопку с галочкой в верхнем левом углу рабочей области программы. После окончания проверки в нижнем левом углу должно быть написано: «Компиляция завершена». Если же нижняя часть программы стала красной, значит, где-то вкралась ошибка. Для начала попробуй скачать заново исходные коды для этого набора и открыть эту же программу. Если все заработало, то, скорее всего, ты случайно набрал какой-то ненужный символ. Если же ты и сейчас не можешь скомпилировать программу, пиши нам в службу поддержки.



- Действие 7.** Загрузи программу в микрокомпьютер. Для этого надо нажать на круглую зеленую кнопку *Вгрузить*, которая находится справа от кнопки компиляции скетча в Действии 6. В узком черном окне внизу через некоторое время должно высветиться *Transfer complete*. Если появится надпись *Transfer incomplete*, а все предыдущие шаги ты сделал верно, смотри [Главу 13](#), пункт 13.3.
- Результат** Маленький светодиод на плате моргает зеленым цветом с частотой раз в секунду! Если программа зависла, нажми на плате кнопку *Перезагрузка скетча Arduino (SHLD RST)*.
- Не работает?** Загрузи заново программу.

СХЕМА 2 ПОСТОЯННОЕ СВЕЧЕНИЕ СВЕТОДИОДА

После долгих приготовлений настала пора взяться за создание более серьезных вещей. Предлагаю тебе создать свою первую электрическую цепь. Пока без программирования, но для начала оно и не нужно.

Собирая эту схему, мы изучим основы работы с макетной платой и подадим питание с платы Intel® Edison на внешний светодиод, чтобы он загорелся.

- Что необходимо** Беспаяная макетная плата. Во всех остальных схемах она будет необходима по умолчанию, и далее я об этом писать не буду.

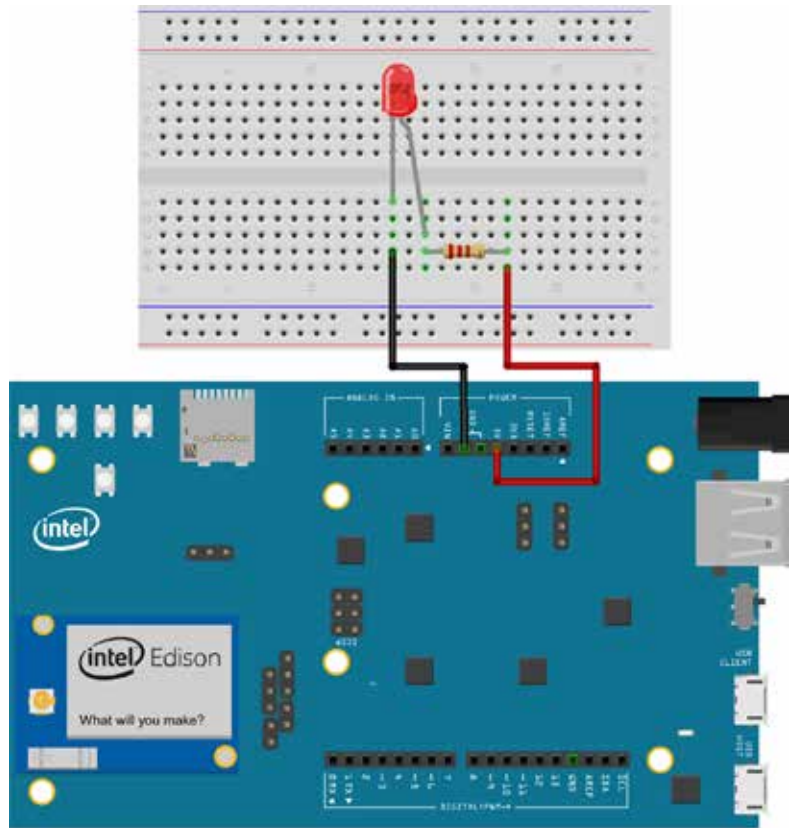
Два провода «папа-папа» (с двух сторон штырьки, такие провода еще называют «штекер-штекер»). Цвет не важен. Во всех схемах, особенно сложных, провода, которые подсоединены к земле (пин GND), должны иметь черный, коричневый, темно-зеленый или синий цвет. Провода, которые подсоединены к питанию (пины VCC, 5 В, 3.3 В), должны иметь красный или оранжевый цвет. Сигнальные провода, по которым будут идти управляющие сигналы (цифровые и аналоговые пины), должны иметь белый, желтый или оранжевый цвет.

Любой одноцветный светодиод (все одноцветные светодиоды имеют только два вывода).

Резистор на 220 Ом (красный-красный-коричневый). Будет отлично, если ты согласишься посмотреть таблицу цветовых кодировок резисторов ([Глава 2](#)), чтобы понять, почему все именно так, как я сказал.

Кабель микро-USB.

Схема 2



Сборка схемы

Начальные приготовления. Убедись, что на плату не подается напряжение, т. е. все микро-USB-кабели отсоединены от платы или от компьютера. Внимательно изучи Схему 2. Обрати внимание на подключения проводов.

Действие 1.

Установи светодиод выводами на макетную плату вниз. Поставь светодиод перпендикулярно макетной плате и воткни его выводы в отверстия на макетной плате. Ножки светодиода войдут в макетку приблизительно на 5 мм. Не стоит сильно давить, чтобы не погнуть ножки светодиода и не повредить его. Это должно выглядеть приблизительно как на картинке ниже.

Действие 2.

Теперь возьми провод (в моем случае он черный) и под соедини его, как показано на схеме, к короткому выводу светодиода. То есть установи в отверстие, которое находится в том же столбце, что и короткий вывод светодиода.



- Действие 3.** Возьми свободный вывод черного провода и подсоеди- ни в разъем, возле которого написано GND.
- Действие 4.** Теперь резистор. Как видишь, он прямой. Для того чтобы поставить его на макетную плату, согни его буквой «П», оставив приблизительно по 3 мм прямой проволоочки с каждой стороны от самого «цилиндра» резистора. И уста- нови его, как показано на Схеме 2: один вывод поставь в отверстие под длинной ножкой светодиода, а второй – на той же «строке» отверстий, но правее настолько, на сколь- ко хватает длины резистора до отверстия.
- Действие 5.** Возьми второй провод (на схеме красный провод) и под соедини его в столбец, где стоит второй вывод ре- зистора.
- Действие 6.** Свободный конец второго провода под соедини к пину 5 В.
Перед тем как приступить к Схеме 3, разбери Схему 2.
- Результат** Ты подал питание на светодиод при помощи макетной платы и платы Intel® Edison. Это слишком просто. Давай добавим к этой схеме немного программирования!
- Не работает?** Проверь соединения согласно Схеме 2.
Отсоедини и под соедини снова все элементы электри- ческой цепи на макетной плате.
Поменяй светодиод на другой.

СХЕМА 3 МОРГАНИЕ ВНЕШНИМ СВЕТОДИОДОМ

Настало время объединить первую и вторую схемы в единое целое!

Мы изучим основы коммутации устройств на макетной плате и будем моргать внешним светодиодом на плате Intel® Edison С программы, а также выведем текстовую информацию о состоянии светодиода через Serial-порт.

Что необходимо Два провода «папа-папа» (с двух сторон штырьки). Можешь взять их из предыдущей схемы.

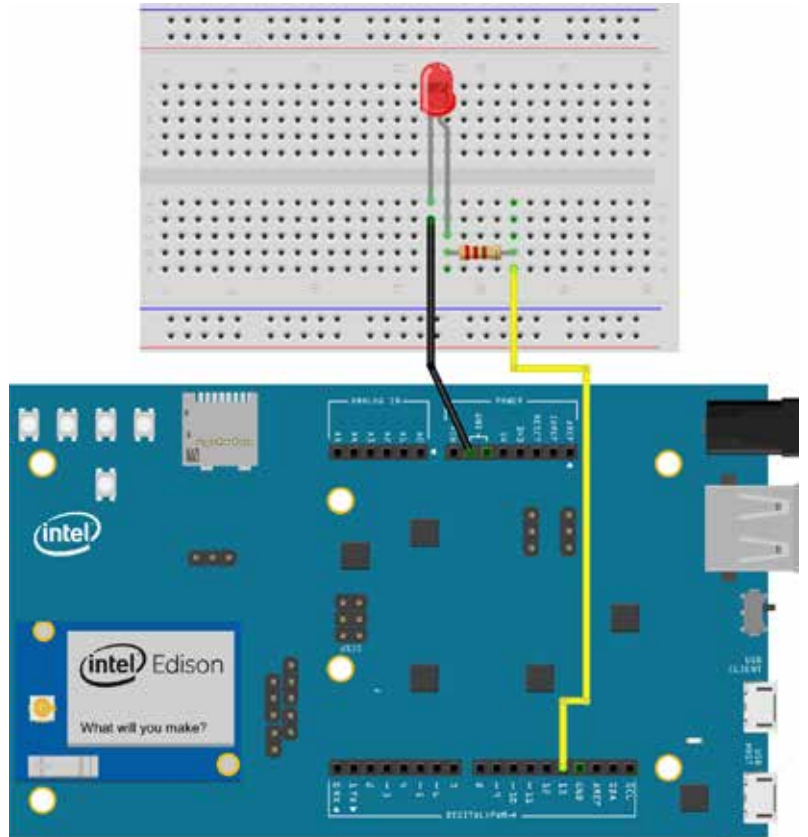
Любой одноцветный светодиод. Возьми, к примеру, теперь зеленый.

Два кабеля микро-USB из набора.

Резистор на 220 Ом.

Действия будут во многом похожи на сборку Схемы 2, за исключением одного небольшого изменения.

Схема 3



Сборка схемы

Начальные приготовления. Убедись, что на плату не подается напряжение, т. е. все микро-USB-кабели отсоединены от платы или от компьютера. Внимательно изучи Схему 3. Обрати внимание на подключения проводов.

Действие 1.

Провод, который подсоединен к пину 5 В, тебе надо отсоединить и подсоединить к цифровому пину под номером 13.

Действие 2.

Теперь подключи микро-USB-кабель к OTG-порту платы Intel® Edison (микро-USB посередине платы, смотри схему подключений Intel® Edison) и подключи кабель к компьютеру. После того как ты подключишь питание к плате, начнет выполняться скетч из Схемы 1 и светодиод, скорее всего, начнет мигать, но ты все равно загрузи необходимый для схемы скетч.

Действие 3.

Запусти Arduino IDE и открой программу A1_A_S3_Blink_led.ino. Прочти все комментарии, которые приведены в тексте программы, чтобы лучше понять, как она работает.

Действие 4.

Перед загрузкой проверь код на ошибки и загрузи программу на плату.

Результат

Внешний светодиод моргает! Если твоя программа зависла, то нажми кнопку на плате Перезагрузка скетча Arduino (SHLD RST).

СХЕМА 4 ПРОТОТИП «УМНОГО» ДОМА НА ОСНОВЕ СВЕТОДИОДА И ФОТОРЕЗИСТОРА

Предлагаю собрать схему включения и выключения светодиода в зависимости от количества падающего на фоторезистор света. Эту схему можно применить в комплексе «умный» дом для экономии электроэнергии. Свет выключился, когда наступил день, и включился, когда наступил вечер.

В схеме будут считываться показания фоторезистора, на основании которых будет приниматься решение о включении или выключении светодиода.

Что необходимо Любой одноцветный светодиод.

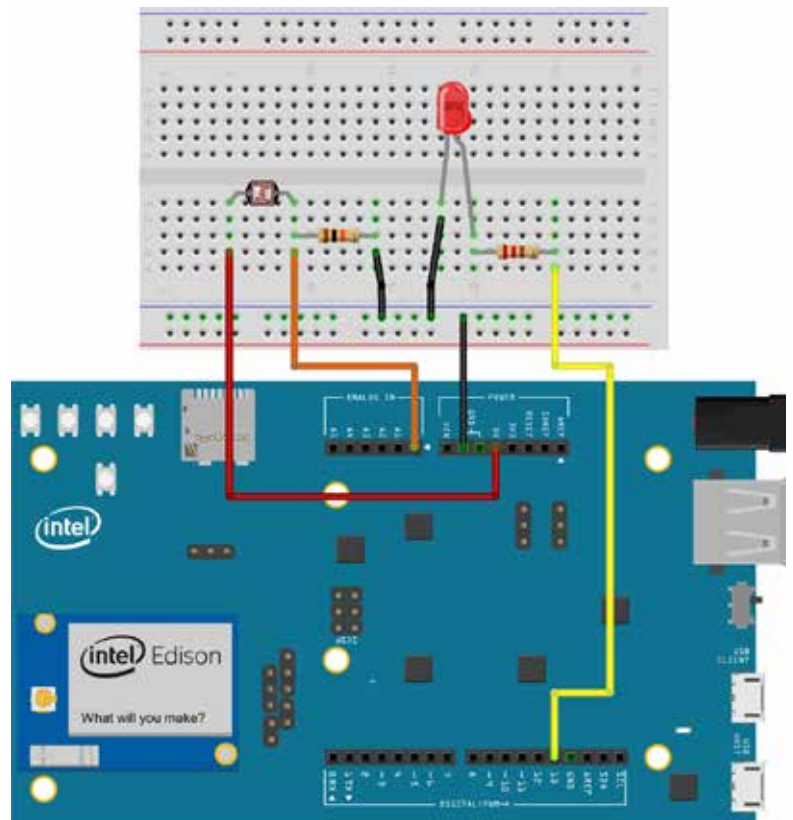
6 проводов «папа-папа».

Фоторезистор.

Резистор на 220 Ом.

Резистор на 10 кОм.

Схема 4



Сборка схемы

Действие 1.

В этой схеме два компонента (фоторезистор и светодиод), у них будет общая «земля». Чтобы не подключать к каждому пину GND Intel® Edison «землю», т. к. в конце концов пинов GND на главной плате не хватит, мы ее выведем на макетную плату, где можно организовать

большое количество нужных выводов. Подключим черный провод к разъему GND на плате Intel® Edison, а второй вывод провода подключим к длинной строке на макетной плате, рядом с которой нарисована длинная синяя линия. Мы вывели «землю» на макетную плату.

- Действие 2.** Установи светодиод на макетную плату.
- Действие 3.** Установи резистор на 220 Ом (вспомни Схему 3).
- Действие 4.** Возьми второй провод, у меня он желтый (сигнальный), и подсоедини в столбец, где стоит второй вывод резистора.
- Действие 5.** Свободный конец желтого провода подсоедини к цифровому пину под номером 13.
- Действие 6.** Подсоедини провод к минусу светодиода, второй конец к «земле» на макетной плате.
- Действие 7.** Установи два вывода фоторезистора в отверстия в разных столбцах.
- Действие 8.** Установи резистор на 10 кОм, как показано на Схеме 4. Вывод резистора должен быть в одном столбце с выводом фоторезистора.
- Действие 9.** Подсоедини один конец данного резистора к «земле».
- Действие 10.** Теперь подсоедини провод к столбцу, в котором вывод фоторезистора и резистора. Второй конец подсоедини к аналоговому пину A0 на плате Intel® Edison.
- Действие 11.** Оставшийся свободный контакт от фоторезистора подключи к пину питания 5 В на плате.
- Действие 12.** Проверь правильность своей схемы, сверившись со Схемой 4.
- Действие 13.** Теперь подключай микро-USB-кабель к OTG-порту платы Intel® Edison (микро-USB посередине платы, смотри схему подключений Intel® Edison) и подключи кабель к компьютеру.
- Действие 14.** Запусти Arduino IDE и открой программу A1_A_S4_Photoresistor_with_led.ino. Прочти все комментарии, которые приведены в тексте программы, чтобы лучше понять, как она работает.
- Действие 15.** Перед загрузкой проверь код на ошибки и загрузи программу на плату.

Результат Светодиод включается и выключается в зависимости от количества падающего на фоторезистор света. Закрой рукой фоторезистор и светодиод загорится, убери руку, и он потухнет. Делать это желательно в освещенной комнате. Если твоя программа зависла, не реагирует на изменение света, нажми кнопку на плате Перегрузка скетча Arduino (SHLD RST).

Открой монитор последовательного порта, и увидишь результат выполнения программы на своем экране. Когда закрываешь фоторезистор рукой, на мониторе отображается Night, а когда открываешь доступ к свету – Day.

Не работает?

Проверь соединения согласно Схеме 4: правильно ли ты все подключил.

Попробуй изменять значения показателя освещенности. Вместо 700 можешь поставить 500, 1000; шаг в 100 оптимальный. Экспериментируй!

Я специально подключил внешний светодиод к 13-му пину. На 13-м пине установлен светодиод, встроенный в плату Intel® Edison. Будем считать, что это наш помощник. Если он моргает, а внешний светодиод нет, значит, проблема в контактах и правильности сборки схемы.

Отсоедини и подсоедини снова все элементы электрической цепи на макетной плате.

Поменяй светодиод на другой.

Загрузи заново программу.

Краткая теория по принципам работы программы

Как я уже писал, фоторезистор – это резистор с переменным сопротивлением. Аналоговый пин A0 настроен на вход. Я также писал, что ты можешь измерить напряжение, которое подается на данный пин, с помощью функции `analogRead()`.

Если функция `analogRead()` выдает значение 0, то напряжение, подаваемое на данный аналоговый пин, соответствует 0 В, если 1024, то 5 В. А так как фоторезистор меняет сопротивление, напряжение, подаваемое на пин A0, от изменения сопротивления также меняется, и, следовательно, значение функции меняется от 0 до 1024. В действительности же значения немного другие: показания, считанные с аналогового порта A0, изменяют значение от ~500 до ~950. Такие значения обусловлены внутренним устройством фоторезистора. Я настроил параметр на определенную величину. Можешь изменять его и смотреть, что из этого выйдет.

А зачем резистор к фоторезистору?

Для снятия показания с фоторезистора необходим делитель напряжения из фоторезистора и резистора на 10 кОм (я упоминал об этом, описывая компоненты). На аналоговый вход A0 считывается потенциал точки делителя — он изменяется пропорционально сопротивлению фоторезистора. Если не устанавливать резистор

на 10 кОм., то потенциал постоянно будет равен нулю и изменений на порту A0 не будет. Прочти про делитель напряжения здесь: <http://tiny.cc/a1-voltage-divider>.

Можешь провести такой эксперимент

Загрузи стандартную программу Схемы 4. Проверь, чтобы Схема 4 работала как надо. Не выключая питания от платы, вытащи провод из пина A0. Теперь отсоедини провод питания от фоторезистора на макетной плате и подключи его в пин A0. Светодиод потух! Отсоедини, светодиод загорится. Вот практическое подтверждение моим словам. Функция `analogRead()` в программе считывает значение напряжения на выводе A0. Когда ты подаешь 5 В на A0, значение напряжения на выводе становится высоким. Если отключаешь – низким. Весь наш мир аналоговый, и очень важно уметь работать с аналоговыми устройствами.

Схема 4_1 для любознательных

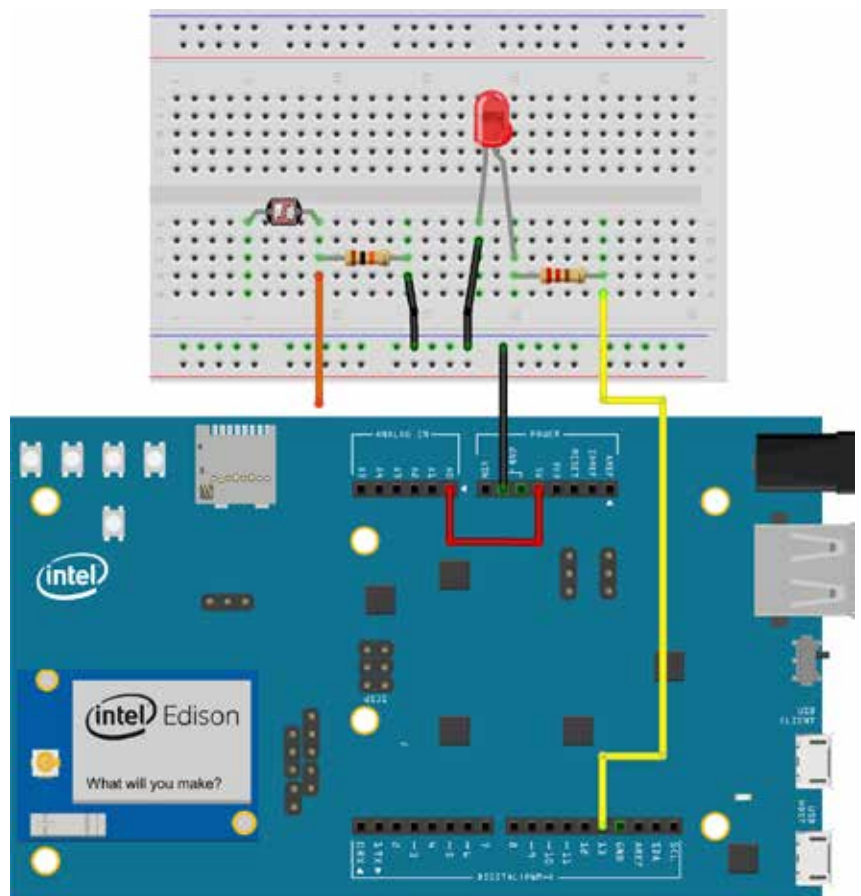


СХЕМА 5 ВКЛЮЧЕНИЕ И ВЫКЛЮЧЕНИЕ СВЕТОДИОДА С ПОМОЩЬЮ КНОПКИ

Теперь мы научим микрокомпьютер понимать окружающий мир через физическое взаимодействие, а именно через кнопку. Кнопки окружают нас повсюду: в домах, автомобилях, смартфонах, часах и еще много-много где! Кнопка – это устройство, которое помогает настраивать различные параметры, устройство очень немудреное: она просто размыкает или замыкает электрическую цепь в нужных местах.

В данной схеме нажатие на кнопку будет приводить к обнулению напряжения (LOW) на входе, включению светодиода и отображению на мониторе последовательного порта текста Light is ON, а при отжатой кнопке напряжение будет высоким (HIGH), светодиод выключится, а на мониторе последовательного порта появится текст Light is OFF.

Что необходимо Одноцветный светодиод.

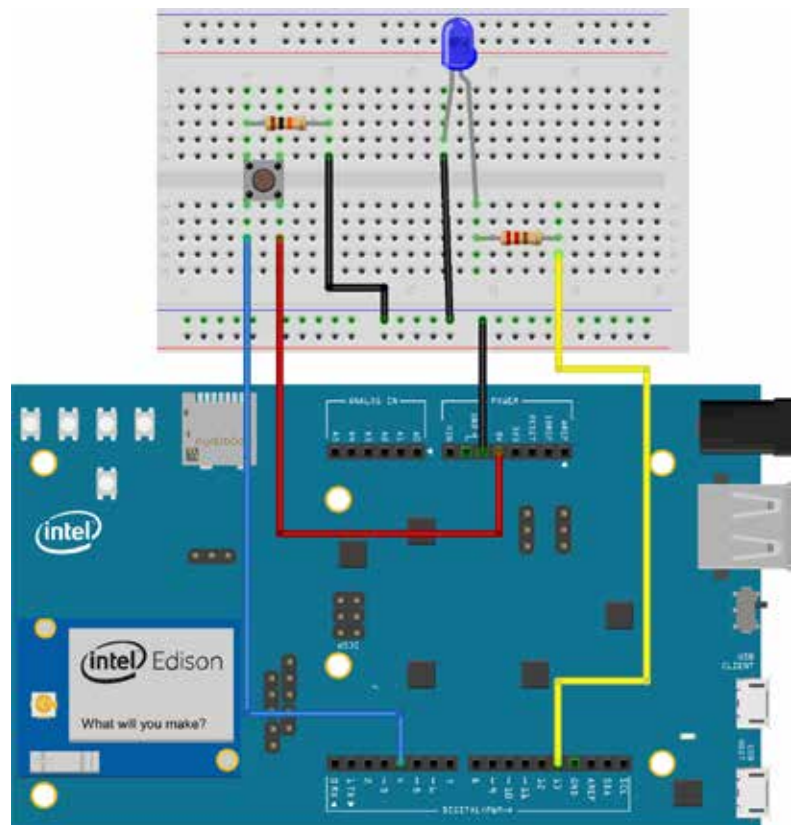
Шесть проводов «папа-папа».

Одна тактовая кнопка.

Резистор на 220 Ом.

Резистор на 10 кОм.

Схема 5



Сборка схемы:

- Действие 1.** Подключим черный провод к разъему GND на плате Intel® Edison, а второй конец – к длинной строке на макетной плате, рядом с которой нарисована длинная синяя линия. Мы вывели «землю» на макетную плату.
- Действие 2.** Подключим красный провод к разъему 5 В на плате Intel® Edison, а второй конец – к длинной строке на макетной плате, рядом с которой нарисована длинная красная линия. Теперь мы вывели питание 5 В на макетную плату.
- Действие 3.** Установи светодиод на макетную плату.
- Действие 4.** Установи резистор на 220 Ом.
- Действие 5.** Возьми второй провод и под соедини его в столбец, где стоит второй вывод резистора.
- Действие 6.** Свободный конец желтого провода под соедини к цифровому пину под номером 13.
- Действие 7.** Под соедини провод к минусу светодиода, второй конец – к «земле» на макетной плате.
- Действие 8.** Установи кнопку как показано на Схеме 5, т. е. посередине макетной платы.
- Действие 9.** Установи резистор на 10 кОм, как показано на Схеме 5. Первый вывод резистора должен быть в одном столбце с верхним левым выводом кнопки.
- Действие 10.** Второй вывод резистора под соедини к «земле».
- Действие 11.** Теперь под соедини провод к столбцу, в котором находится нижний левый вывод кнопки. Вторым концом провода под соедини к цифровому пину 4 на плате Intel® Edison.
- Действие 12.** Нижний правый вывод кнопки под соедини к питанию 5 В на макетной плате.
- Действие 13.** Проверь правильность своей схемы, сверившись со Схемой 5.
- Действие 14.** Теперь под соедини микро-USB-кабель к плате Intel® Edison.
- Действие 15.** Запускай Arduino IDE и открывай программу A1_A_S5_One_button_with_led.ino. Прочти все комментарии, которые приведены в тексте программы, чтобы лучше понять, как она работает.
- Действие 16.** Перед загрузкой проверяй код на ошибки и загружай программу на плату.

Результат

Твой внешний светодиод включается и выключается в зависимости от состояния кнопки. Если кнопка нажата, светодиод горит, не нажата – не горит.

Открой монитор последовательного порта, и увидишь результат выполнения программы на своем экране. На мониторе отображается Led is ON либо Led is OFF.

Не работает?

Проверь соединения согласно Схеме 5.

Я специально подключил внешний светодиод к 13-му пину. На 13-м пине установлен светодиод, встроенный в плату Intel® Edison. Будем считать, что это такой небольшой помощник. Если он загорается при нажатии кнопки и тухнет, когда кнопка отпущена, а внешний светодиод нет, значит, проблема в контактах и правильности сборки схемы.

Перезагрузи плату.

Отсоедини и подсоедини снова все элементы электрической цепи на макетной плате.

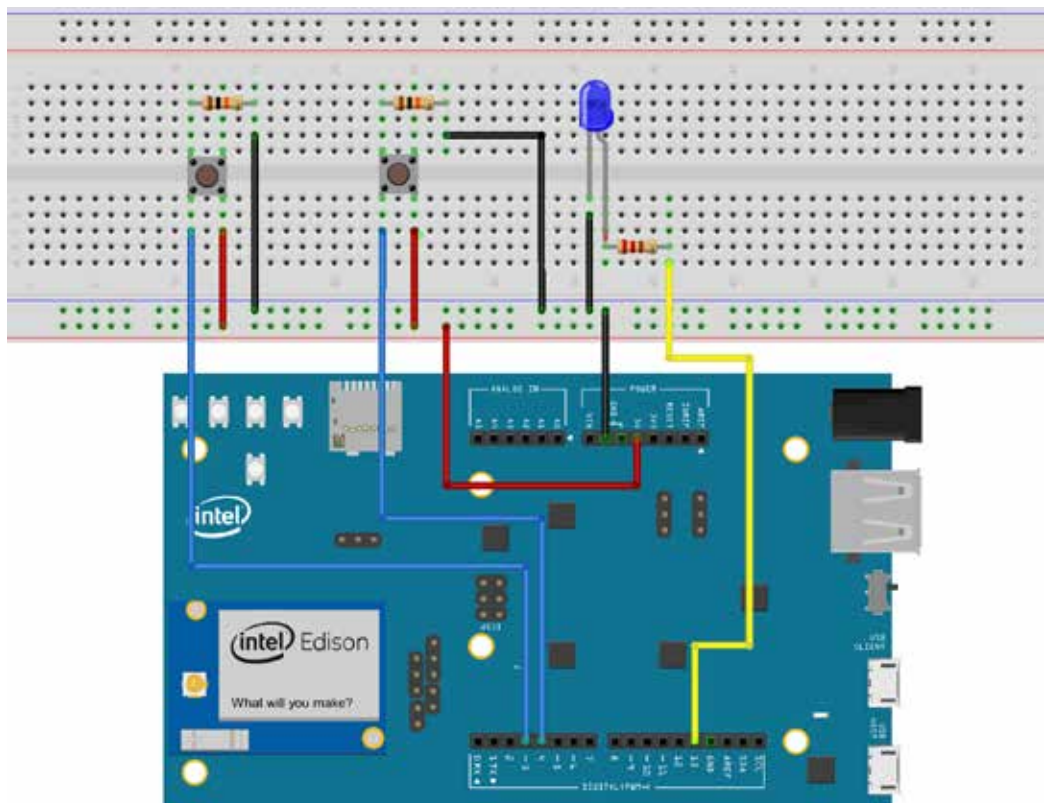
Поменяй светодиод на другой.

Загрузи заново программу.

СХЕМА 5.1 ВКЛЮЧЕНИЕ И ВЫКЛЮЧЕНИЕ СВЕТОДИОДА С ПОМОЩЬЮ ДВУХ КНОПОК

Не разбирай Схему 5. Сейчас мы добавим одну кнопку и усложним программную составляющую.

Схема 5.1



Сборка схемы

Все действия по подключению второй кнопки, абсолютно те же, что и в случае с первой кнопкой в Схеме 5, за исключением Действия 10. Мы подключим вторую кнопку к цифровому пину 3.

Действие 1.

Теперь подсоедини провод к столбцу, в котором находится нижний левый вывод кнопки. Вторым концом провода подсоедини к цифровому пину 3 на плате Intel® Edison.

Действие 2.

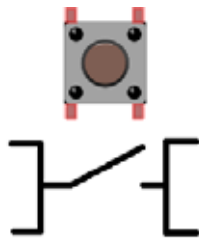
Запусти Arduino IDE и открой программу `A1_A_S5_1_Two_buttons_with_led.ino`.

Результат

Твой внешний светодиод включается и выключается в зависимости от состояния двух кнопок. Одна включает его, а другая выключает.

Краткая теория по использованию кнопки:

На первый взгляд может показаться, что подключить кнопку и работать с ней очень просто. Кнопку с четырьмя контактами стоит рассматривать, как две пары рельс, которые соединяются при нажатии.



При замыкании и размыкании цепи кнопкой между ее пластинами возникают микроискры. Эти электрические искры провоцируют до десятка переключений за несколько миллисекунд. Такое явление называется дребезгом. Обязательно учитывай это, если тебе необходимо фиксировать количество нажатий на кнопку. Наилучший способ – сделать небольшую задержку в 5 миллисекунд при помощи функции `delay()` перед следующим чтением состояния кнопки.

Существует два способа подключения кнопки. Это подключение со стягивающим резистором и с подтягивающим резистором. Такие схемы подключения исключают возможность появления негативных шумов на выводе, с которого ты считываешь состояние кнопки. На рисунке слева кнопка подключена по схеме с подтягивающим резистором, справа – со стягивающим.

Главные различия между двумя способами подключения

В схеме с подтягивающим резистором, когда кнопка нажата (контакты в кнопке замкнуты), напряжение на порту, с которого считывается состояние кнопки, равно 0. Когда не нажата (контакты разомкнуты), напряжение на порту соответствует напряжению питания.

В схеме со стягивающим резистором все наоборот. То есть когда кнопка нажата, то напряжение на порту соответствует напряжению питания. А когда не нажата, то напряжение на порту с которого считывается состояние кнопки равно 0.

Схема подключения кнопок. С подтягивающим резистором (кнопка слева) и со стягивающим резистором (кнопка справа).

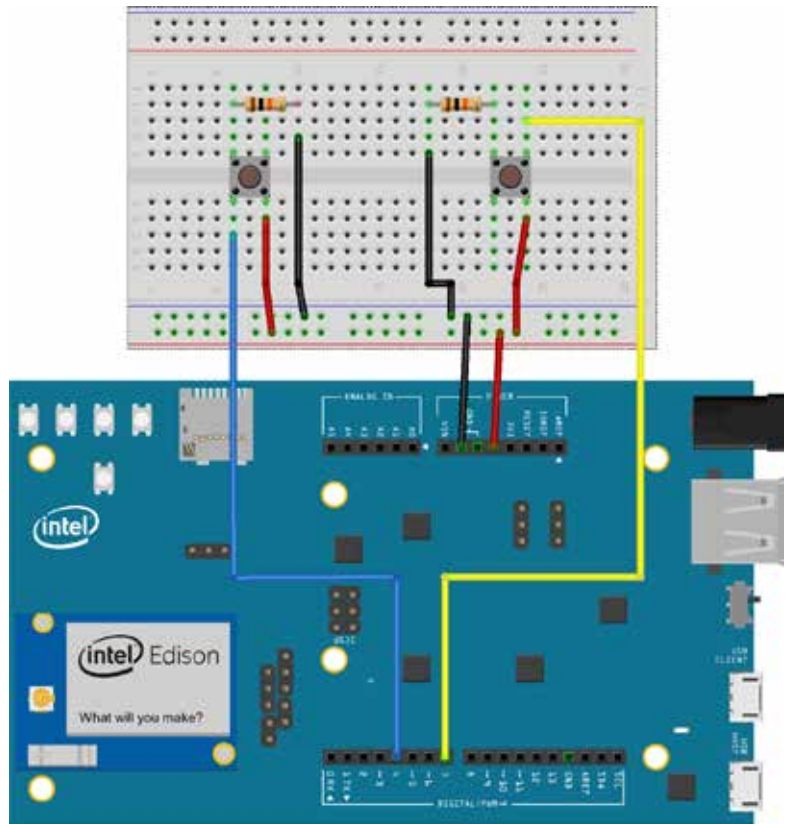


СХЕМА 6 СВЕТОРЕГУЛЯТОР НА ОСНОВЕ ПОТЕНЦИОМЕТРА

Сейчас мы освоим работу с потенциометром. И соберем интересную и довольно полезную в реальной жизни схему. Наверное, многие из вас помимо обычных выключателей видели также диммеры или же светорегуляторы, позволяющие настроить яркость освещения в квартире.

Вот и мы сделаем простейший светорегулятор на основе потенциометра. При изменении угла поворота потенциометра будет меняться и яркость светодиода.

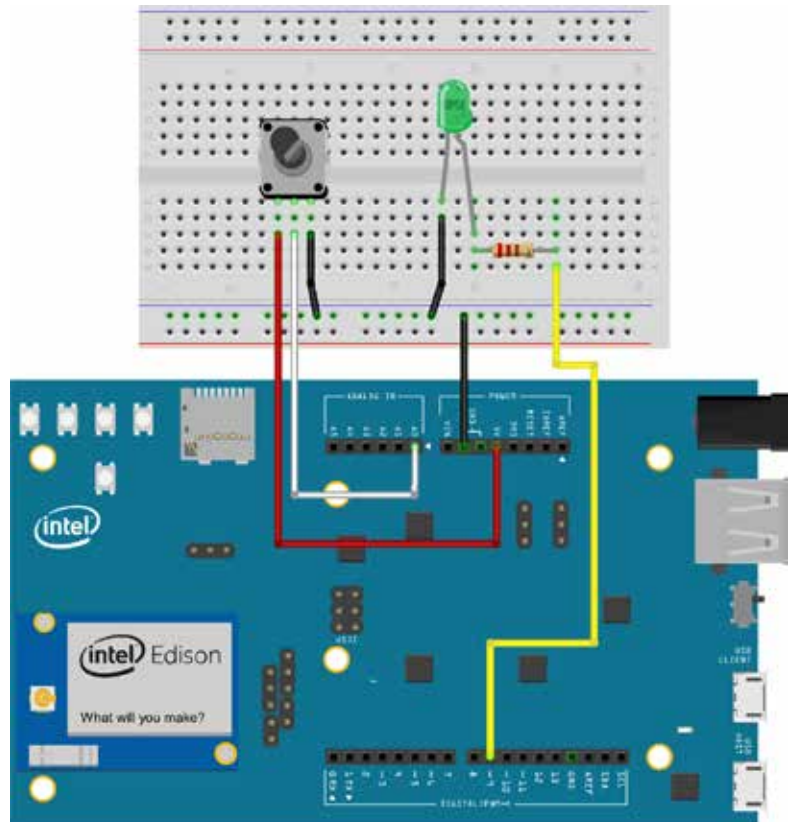
Что необходимо Одноцветный светодиод.

Шесть проводов «папа-папа».

Потенциометр.

Резистор на 220 Ом.

Схема 6



Сборка схемы

Действие 1. Подключи светодиод к основной плате. Действия описаны в Схемах 3, 4, 5. За исключением одной важной детали: светодиод надо подключить с цифровому пину номер 9. Этот вывод поддерживает широтно-импульсную модуляцию благодаря чему светодиод сможет менять яркость.

Действие 2. Поставь потенциометр ручкой регулировки сопротивления (ротором) на себя, а выводами вниз. В таком положении выводы питания – это крайние выводы (их полярность не имеет значения), а центральный вывод будет выдавать измененное значение напряжения.



Действие 3. Подсоедини провод к крайнему левому выводу потенциометра и к пину питания 5 В на плате.

Действие 4. Подсоедини провод к центральному выводу потенциометра и к аналоговому пину 0 на плате.

Действие 5. Подсоедини провод к крайнему правому выводу потенциометра и к «земле» на макетной плате.

Действие 6. Проверь правильность своей схемы, сверившись со Схемой 6.

Действие 7. Теперь подключай микро-USB-кабель к плате Intel® Edison.

- Действие 8.** Запусти Arduino IDE и открой программу A1_A_S6_Pot_and_led.ino.
- Действие 9.** Перед загрузкой проверь код на ошибки и загрузи программу на плату.
- Результат.** Твой внешний светодиод меняет яркость в зависимости от значения сопротивления потенциометра, которое ты выставляешь, вращая ротор потенциометра.
- Не работает?** Проверь соединения согласно Схеме 6.
Светодиод должен быть подключен к 9-му цифровому пину.
Проверь подключение потенциометра.
Перезагрузи плату.
Поменяй светодиод на другой.
Загрузи заново программу.
Отсоедини и подсоеди́ни снова все элементы электрической цепи на макетной плате.

СХЕМА 7

СИНТЕЗАТОР ЗВУКА НА ОСНОВЕ ПЬЕЗОЭЛЕКТРИЧЕСКОГО ЗВОНКА

Эта схема дает возможность освоить работу с пьезоэлектрическим звонком или пьезодинамиком (есть еще названия «баззер» или «зуммер»). Мне кажется, что все видели пищалые детские игрушки и слышали звук, который они издают, или же слышали, как работает будильник наручных часов. За создание звуков в этих и многих других устройствах отвечает пьезодинамик. Предлагаю научиться управлять звуком. Звук – это аналоговое явление и его можно создать при помощи нашей цифровой платы благодаря относительно высокой скорости работы микроконтроллера. Подробнее про пьезоэлектрический эффект почитай здесь: <http://tiny.cc/a1-piezo>.

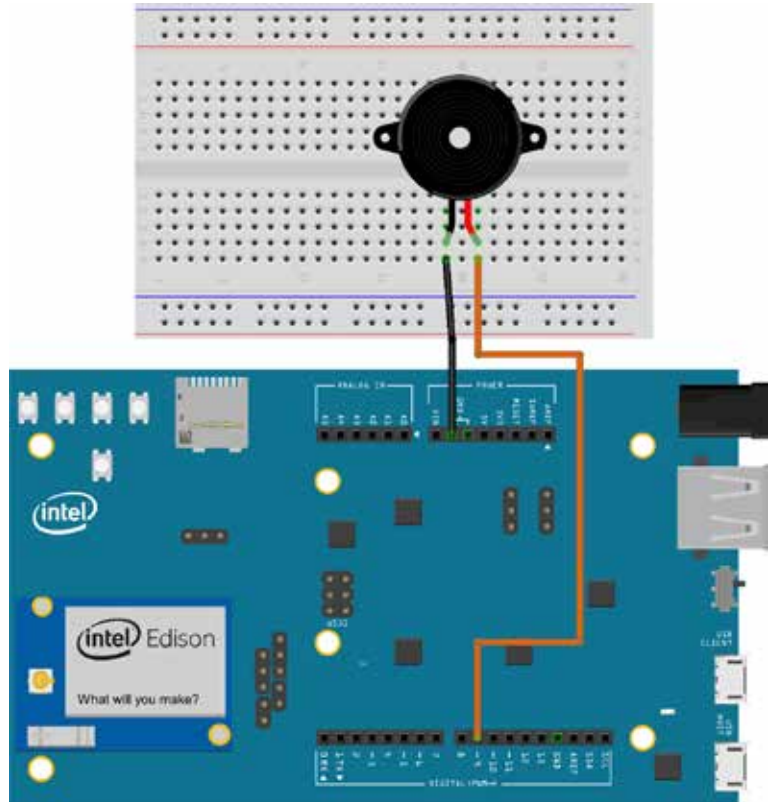
Пьезодинамик щелкает каждый раз, когда на него подается электрический импульс. Если посылать эти импульсы с определенной частотой (например, 440 раз в секунду для воспроизведения ноты ля первой октавы), то щелчки превратятся в музыкальный тон. Предлагаю заставить нашу плату сыграть какую-нибудь мелодию.

Мы сделаем музыкальный инструмент при помощи пьезодинамика и будем извлекать звуки.

Что необходимо Два провода «папа-папа».

Пьезодинамик.

Схема 7



Сборка схемы

- Действие 1.** Установи пьезодинамик штырьками вниз на макетную плату. Сними защитную пленку с верхней части пьезодинамика. В верхней части корпуса над одним из выводов должен стоять знак «+». Это красный вывод баззера на Схеме 7. Соответственно, другой вывод – это «минус».
- Действие 2.** Подсоедини провода от пьезодинамика, как показано на схеме 7.
- Действие 3.** Проверь правильность своей схемы, сверившись со Схемой 7.
- Действие 4.** Теперь подключи микро-USB-кабель к плате Intel® Edison.
- Действие 5.** Запусти Arduino IDE и открой программу A1_A_S7_Sound_buzzer.ino.
- Действие 6.** Перед загрузкой проверь код на ошибки.
- Действие 7.** Загрузи программу на плату.
- Результат** Твой пьезодинамик играет гамму и никогда не ошибается!
- Не работает?** Проверь соединения согласно Схеме 7.

Пьезодинамик должен быть подключен к цифровому пину номер 9.

Поменяй выводы подключения пьезодинамика местами; может, ты просто перепутал «плюс» с «минусом» либо не попал проводами в отверстия, где стоят выводы динамика.

Перезагрузи плату.

Загрузи заново программу.

Отсоедини и подсоедини снова все элементы электрической цепи на макетной плате.

СХЕМА 8 ТЕМПЕРАТУРНЫЙ ДАТЧИК СО СВЕТОВОЙ СИГНАЛИЗАЦИЕЙ

Предлагаю сделать собственный электронный термометр на базе платы Intel® Edison и термистора. Это очень интересная схема, т. к. данный принцип работы используется во многих отраслях человеческой деятельности. Например, датчик температуры двигателя в автомобиле при повышении температуры будет сигнализировать водителю, что двигатель необходимо охладить, а в системе «умного» дома он используется для определения температуры в комнатах и т. д.

Наша схема позволит измерять температуру с помощью сенсора и выводить данные значения на монитор компьютера. При превышении температуры 29 градусов по Цельсию будет загораться светодиод. Для того чтобы превысить данный порог, достаточно аккуратно взять пальцем термистор и подождать некоторое время, чтобы он нагрелся от пальца.

Что необходимо Термистор.

Резистор на 10 кОм.

Красный светодиод.

Резистор на 220 Ом.

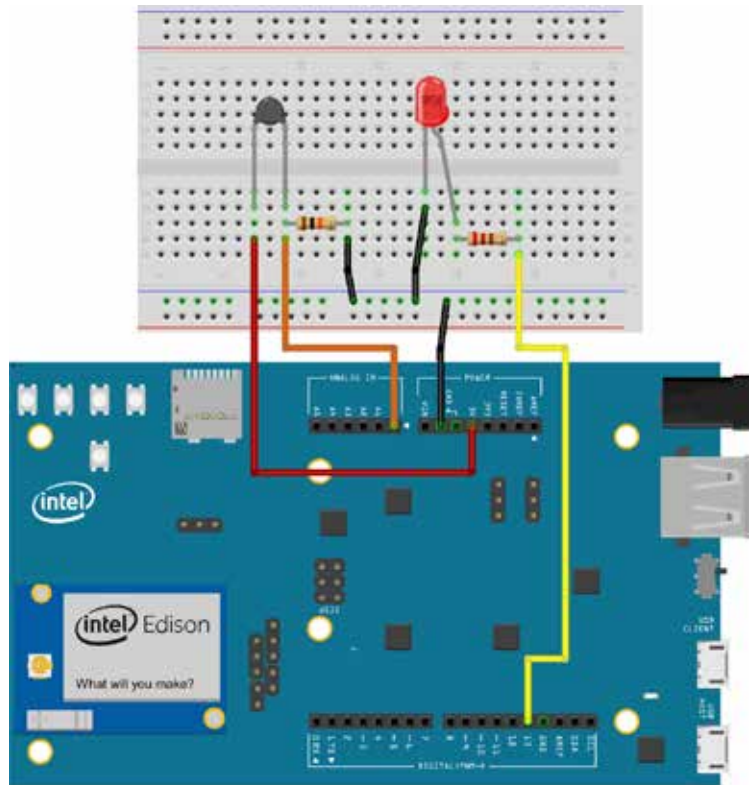
Шесть проводов «папа-папа».

Сборка схемы Если присмотреться, Схема 8 во многом похожа на Схему 4, поэтому действия по сборке Схемы 8 будут идентичны действиям со Схемой 4. Вернись к Схеме 4, чтобы собрать Схему 8. За исключением одной маленькой детали: в Действии 7 вместо фоторезистора тыставишь термистор. Выполнив все действия до Действия 13 включительно, возвращайся сюда.

Действие 1. Запусти Arduino IDE и открой программу A1_A_S8_Thermistor_alarm.ino.

Действие 2. Перед загрузкой проверь код на ошибки и загрузи программу на плату.

Схема 8



Результат

```

Temperature is 22.79
Temperature is 22.63
Temperature is 22.63
Temperature is 22.63
Temperature is 22.55
Temperature is 23.59
Temperature is 26.34
Temperature is 27.57
Temperature is 28.23
Temperature is 28.57
Temperature is 28.65
Temperature is 28.90
Temperature is 29.32
Temperature is 29.66
Temperature is 29.83
    
```

Открой монитор последовательного порта, и увидишь значение температуры, которое изменяется с частотой раз в секунду. Значение `delay()` можно менять, экспериментировать с этим. Когда ты держишь термистор пальцами, светодиод загорается, т. к. температура твоего тела больше 29 градусов по Цельсию.

Пояснения

Директивы для подключения библиотек `#include` включаются в начало программы.

В этой схеме ты подключаешь библиотеку `math.h` для того, чтобы использовать функцию взятия натурального логарифма $\log(x)$. Библиотеки содержат готовые функции, методы, классы, которые можно использовать для своих целей. Их пишут программисты для реализации определенных задач, и многие из таких библиотек можно брать для решения своей задачи, если она похожа на ту, что описана в готовой библиотеке. В случае с библиотекой `math.h` мы просто используем уже готовое решение, созданное до нас. Также библиотеки сокращают размер твоей программы, ведь если у тебя довольно сложная программа, ее можно упростить, отправив часть кода в библиотеку, т. е. создав свою собственную библиотеку с определенным набором функций!

Показания термистора связаны с температурой нелинейно, поэтому нам приходится использовать такую громоздкую формулу.

Не работает?

Проверь соединения согласно Схеме 8.

Если не загорается светодиод, попробуй поменять значения температуры срабатывания в одну и другую сторону с некоторым шагом. Например, поставь сначала 21, потом 29. Должно заработать. Пробуй! Возможно, температура окружающей среды немного иная.

Попробуй догадаться сам, исходя из опыта, который ты приобрел, собирая предыдущие схемы.

СХЕМА 9 СОЗДАНИЕ МУЗЫКАЛЬНОГО ИНСТРУМЕНТА

А теперь немного развлечемся и сделаем имитатор музыкального инструмента терменвокс. Подробнее про этот инструмент можно прочитать здесь: <http://tiny.cc/a1-music>.

Естественно, чтобы создать настоящий терменвокс, надо собрать другую схему. Однако смысл останется прежним. Управлять звуком мы будем бесконтактным путем.

В данной схеме мы будем изменять высоту звука пьезодинамика с помощью фоторезистора, изменяя освещенность рукой.

Что необходимо Фоторезистор.

Резистор на 10 кОм.

Пьезодинамик.

Шесть проводов «папа-папа».

Сборка схемы

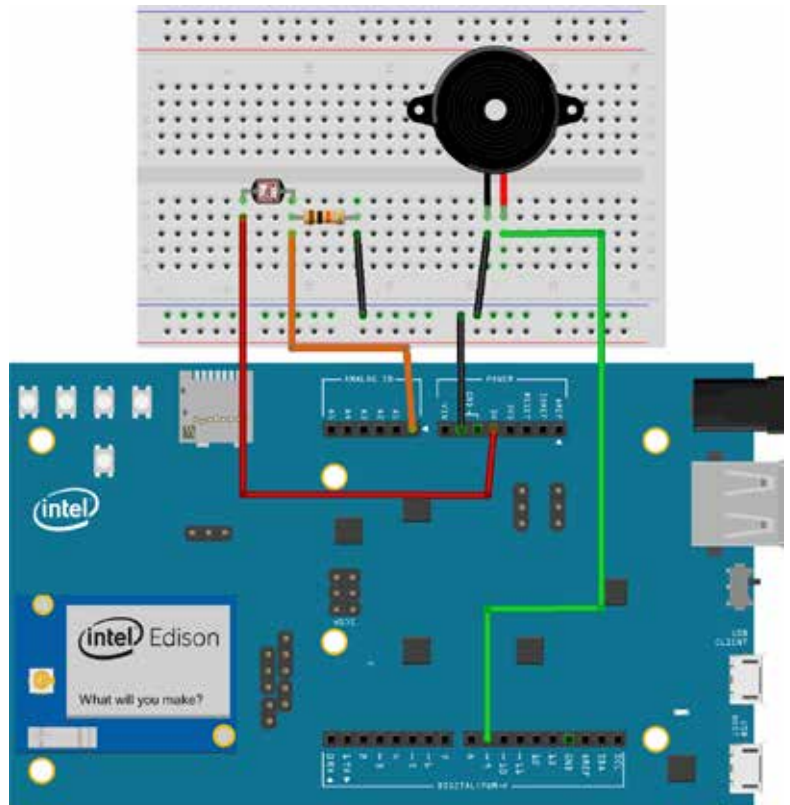
Схема 9 во многом похожа на Схемы 4 и 7. Доверяю тебе собрать данную схему самостоятельно.

После того как ты соединишь все компоненты Схемы 9, выполни следующие действия

Действие 1. Запусти Arduino IDE и открой программу A1_A_S9_Theremin.ino.

Действие 2. Перед загрузкой проверь код на ошибки и загрузи программу на плату.

Схема 9



Результат

С помощью фоторезистора можно изменять высоту звука, издаваемого пищалкой (пьезоэлементом), тем самым можно создавать мелодию!

Пояснения

Функция `map (value, fromLow, fromHigh, toLow, toHigh)` возвращает значение (целочисленное) из интервала `[toLow, toHigh]`. Значение является пропорциональным отображением содержимого в переменной `value` из интервала `[fromLow, fromHigh]`. Это функция пропорционального управления.

Верхние границы функции `map` не всегда бывают больше нижних. Границы могут быть отрицательными. К примеру, значение из интервала `[1, 10]` можно отобразить в интервал `[10, -5]`.

Если при вычислении значения `map` образуется дробное число, оно будет отброшено, а не округлено.

Функция `map` не отбрасывает значения за пределами указанных диапазонов и масштабирует их по заданному правилу.

Если тебе нужно ограничить множество допустимых значений, используй функцию `constrain (value, from, to)`, которая вернет:

`value`, если это значение попадает в диапазон `[from, to]`

`from`, если `value` меньше него

to, если value больше него.

Функция `tone (pin, frequency, duration)` используется для того, чтобы заставить пьезодинамик, который подключен к выводу (`pin`), звучать с частотой (`frequency`) и определенной длительностью (`duration`).

Параметр `duration` не является обязательным для функции `tone`. Если он не будет передан, то звук включится навсегда. Для того чтобы выключить звук, тебе понадобится функция `noTone (pin)`. Она получает номер вывода (`pin`), который нужно выключить.

Не работает?

Проверь соединения согласно Схеме 9.

Попробуй догадаться сам, исходя из опыта, который ты приобрел, собирая предыдущие схемы.

СХЕМА 10 МНОГОЦВЕТНЫЙ RGB-СВЕТОДИОД

Ты уже умеешь работать с обычными одноцветными светодиодами, знаешь, как моргать ими и изменять их яркость. Теперь я предлагаю научиться работать с многоцветным светодиодом, который совмещает в себе все цвета обычных одноцветных светодиодов. Если тебе надо сделать цветовую индикацию о состоянии устройства или работа, то лучше всего использовать многоцветный светодиод, т. к. он экономит место и упрощает всю схему. Ведь проще подключить один многоцветный светодиод, чем устанавливать восемь штук одноцветных, пусть и разных цветов. Во-первых, у тебя останется больше свободных выводов на плате Intel® Edison для подключения дополнительных устройств, а во-вторых, меньше проводов – проще и надежнее электрическая схема!

RGB-светодиод представляет собой три светодиода в одном корпусе: R – красный, G – зеленый и B – синий. Используя различные комбинации этих трех цветов, можно синтезировать практически любой цвет. Тот же принцип используется в мониторах компьютеров, телевизорах и огромных рекламных ЖК-панелях на улицах. Если очень близко подойти к монитору телевизора (или даже взять увеличительное стекло) можно увидеть, что пиксель состоит из красного, синего и зеленого цветов, которые смешивают в разных пропорциях, а твой глаз воспринимает данную смесь как один цвет.

Подробнее про цветовую модель RGB можешь почитать здесь: <http://tiny.cc/a1-rgb>.

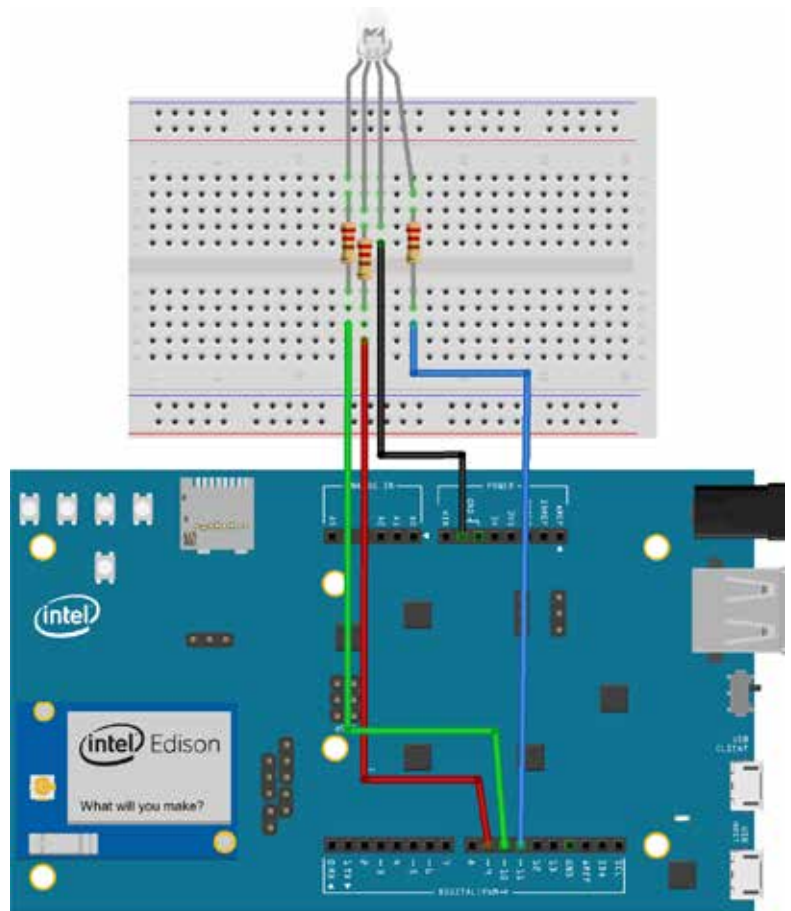
На этот раз мы будем изменять цвет свечения многоцветного светодиода.

Что необходимо Многоцветный светодиод. (Отличается от остальных светодиодов тем, что имеет четыре вывода-ножки.) Многоцветный светодиод может быть с общим анодом (плюсом) либо с общим катодом (минусом) – это самая длинная ножка. В наборе поставляется светодиод с общим катодом.

Три резистора на 220 Ом.

Четыре провода «папа-папа».

Схема 10

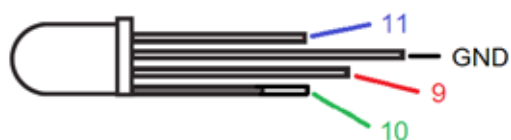


Сборка схемы

Действие 1.

Если присмотреться к светодиоду, то можно увидеть, что его выводы (ножки) разной длины.

На рисунке ниже показано, к каким цифровым выводам на плате Intel® Edison должны быть подключены ножки в зависимости от своей длины.



- Действие 2.** Собери электрическую схему согласно Схеме 10 и рисунку из Действия 1.
- Действие 3.** Запусти Arduino IDE и открой программу A1_A_S10_Rgb_led.ino.
- Действие 4.** Перед загрузкой проверь код на ошибки и загрузи программу на плату.
- Результат** Светодиод меняет цвета.
- Пояснения** В данной программе большое внимание уделено использованию массивов и работе с ними. Использование массивов сильно упрощает написание программы. Подробнее про массивы можно прочесть тут: <http://tiny.cc/a1-arrays>.
- Остальные пояснения к программе находятся в ней самой.
- Не работает?** Проверь соединения согласно Схеме 10.
- Возможно, ты ошибся с подключением выводов светодиодов к портам платы. Проверь все согласно Схеме 10 и рисунку.
- Если программа зависла, перезагрузи плату.
- Попробуй догадаться сам, исходя из опыта, который ты приобрел, собирая предыдущие схемы.

СХЕМА 11 ИГРА «КНОПОЧНЫЕ ГОНКИ»

Предлагаю немного развлечься и собрать свою электронную игру! Цель игры – нажать на кнопку больше раз, чем твой соперник. Игра начинается с красного сигнального светодиода. Как только он потухнет, гонка началась! Вся фишка в том, что светодиод может потухнуть абсолютно неожиданно, и кто быстрее среагирует на это, тот сможет нажать на кнопку большее количество раз.

В данной программе реализована игра «Кнопочные гонки». Она начинается с неожиданного выключения красного светодиода, после чего игроки как можно быстрее нажимают на свои кнопки. Каждое нажатие на кнопку добавляет игроку десять очков. С каждым нажатием светодиоды начинают гореть все ярче, игра продолжается до тех пор, пока кто-либо не наберет больше 255 очков. Как только это произойдет, светодиод победителя начнет моргать. За ходом игры и количеством набранных очков можно интерактивно следить при помощи монитора последовательного порта. В процессе «гонок» там отображается количество очков у каждого игрока, а в конце игры появляется поздравление

победителю. После чего игра автоматически начинается заново.

Что необходимо: Два зеленых светодиода.

Один красный светодиод.

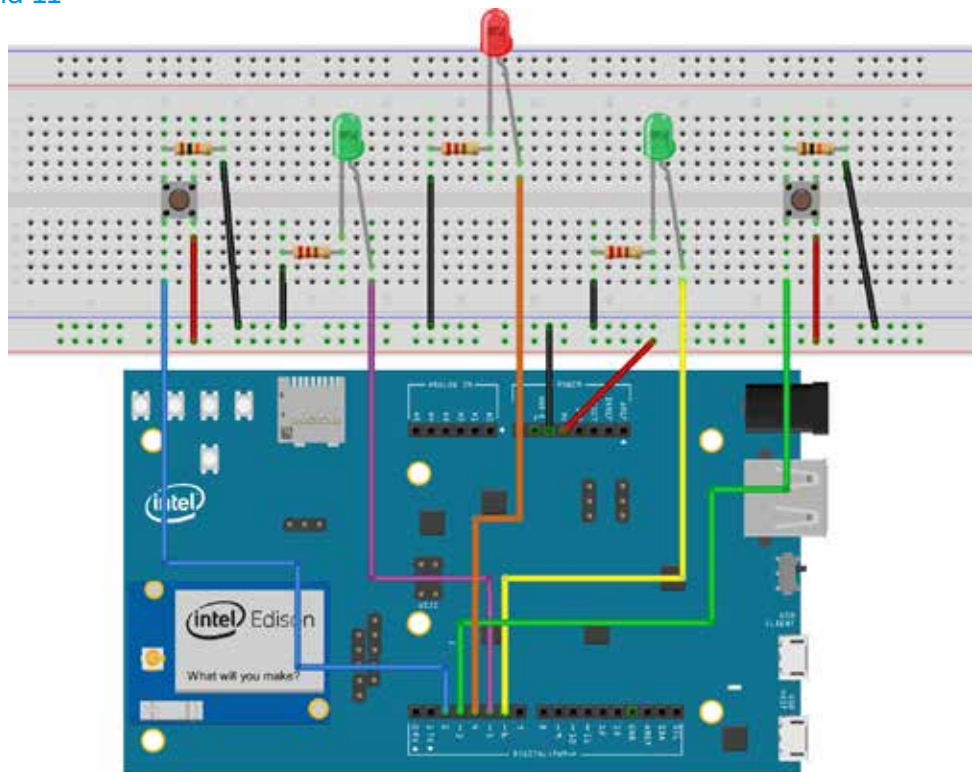
Три резистора на 220 Ом.

Две тактовые кнопки.

Два резистора на 10 кОм.

Четырнадцать проводов «папа-папа».

Схема 11



Сборка схемы

Действие 1. Собери Схему 11.

Действие 2. Запусти Arduino IDE и открой программу A1_A_S10_Rgb_led.ino.

Действие 3. Перед загрузкой проверь код на ошибки и загрузи программу на плату.

Результат Ты можешь играть в собственную игру!

Не работает? Проверь соединения согласно Схеме 11.

Возможно, ты ошибся с подключением выводов светодиодов к портам платы.

Если программа «зависла», перезагрузи плату.

Попробуй догадаться сам, исходя из опыта, который ты приобрел, собирая предыдущие схемы.

СХЕМА 12 ВКЛЮЧЕНИЕ И ВЫКЛЮЧЕНИЕ СВЕТОДИОДА С ПОМОЩЬЮ ВЕБ-СЕРВЕРА

Прежде чем приступать к созданию данной схемы, рекомендую тебе ознакомиться с тем, что такое веб-сервер, и прочитать про Wi-Fi по ссылке: <http://tiny.cc/a1-wifi>.

Микрокомпьютер Intel® Edison имеет встроенный модуль Wi-Fi, благодаря которому ты сможешь использовать возможности Wi-Fi в своих будущих проектах. Например, управлять роботом удаленно. Используя Схему 12, управление платой можно осуществлять только локально в пределах действия одной сети Wi-Fi. Как управлять платой Intel® Edison из любой точки мира, я расскажу, когда дело дойдет до В-схем.

Твой рабочий компьютер должен быть в той же сети Wi-Fi, к которой ты собираешься подключать Intel® Edison, SSID сети не должен быть скрыт. Некоторые публичные сети могут иметь портал авторизации. Там необходимо выполнить вход в сеть, прежде чем получить к ней полный доступ. Такие сети не подходят для подключения платы и выполнения нашей задачи. Поэтому нужно использовать сеть без портала авторизации.

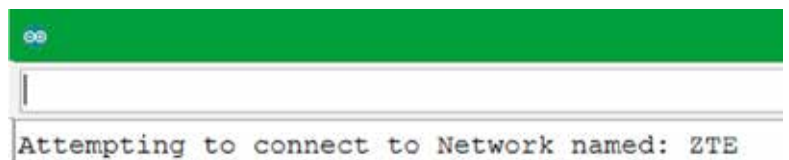
При сборке этой схемы ты изучишь работу веб-сервера, а также научишься управлять встроенным светодиодом на плате Intel® Edison со страницы веб-браузера.

Что необходимо Плата Intel® Edison.

Один кабель микро-USB.

Сборка схемы

- Действие 1.** Запусти Arduino IDE и открой программу A1_A_S12_Web_server_Blink.ino.
- Действие 2.** Теперь тебе надо установить в программе A1_A_S12_Web_server_Blink.ino имя своей сети и пароль. Часть кода, где это необходимо сделать, выделена. Поэтому ты не ошибешься. Больше ничего в программе не трогай.
- Действие 3.** Внимательно прочитай пояснения к программе и загрузи ее на Intel® Edison.
- Действие 4.** Открой монитор последовательного порта в Arduino IDE.
- Действие 5.** Через несколько секунд появится строка с именем сети, к которой пытается подключиться Intel® Edison.



Действие 6. Подожди пару минут. Плата должна подключиться к сети, о чем она известит тебя следующей информацией с именем, выданным IP-адресом платы, уровнем сигнала.

```

Attempting to connect to Network named: ZTE
SSID: ZTE
IP Address: 192.168. 1. 4
signal strength (RSSI): -60 dBm
To see this page in action, open a browser to http://192.168. 1. 4

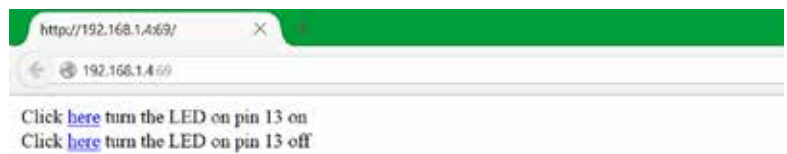
```

Действие 7. Открой браузер и введи адрес платы, который ты получил, выполняя Действие 6. Набери в адресной строке: 192.168.1.4:69

Убедись, что в адресе между числами и точками нет пробелов.

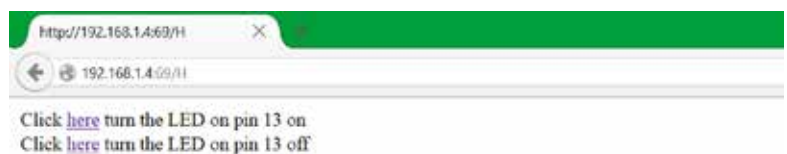


Действие 8. Нажми Enter. Начнет загружаться страница. Подожди 1–2 минуты. Потом должна появиться такая страница с двумя ссылками для включения и выключения светодиода.



Результат.

Кликни по первой ссылке, и встроенный светодиод на плате загорится. Адрес ссылки в браузере изменится.



Кликни по второй ссылке, и светодиод потухнет. Адрес ссылки в браузере также изменится.



Не работает?

Проверь имя сети и пароль, который ввел в Arduino-скетче.

Открыв монитор последовательного порта, нажми на кнопку перезагрузки Arduino-скетча на плате.

8.2. Схемы уровня сложности А+

Дополнительные схемы, комплектующие к которым отсутствуют в базовом наборе А1.

СХЕМА 13 ЭЛЕКТРОМОТОР, ТРАНЗИСТОР И ДИОД

Электромоторы окружают нас повсюду. Поезда в метро, трамваи, троллейбусы, автомобили, подъемные лебедки, строительные краны, сервоприводы и многие другие машины – все они используют электромотор в качестве главного двигателя. Электромоторы бывают с питанием от постоянного либо переменного тока и делятся на коллекторные и бесколлекторные (<http://tiny.cc/a1-motors>).

Электромотором, как и любым другим двигателем, необходимо управлять. И если в некоторых машинах управление происходит через подачу и отключение электроэнергии от мотора, то для управления электромотором в современном автомобиле требуется контроллер, который в паре с регулятором скорости (ESC) меняет параметры скорости вращения электромотора, а также плавность его работы (<http://tiny.cc/a1-esc>).

Порты ввода-вывода Intel® Edison хорошо подходят для управления чем-то маленьким с небольшим потреблением электроэнергии (светодиод). Для управления большими нагрузками (например, электромоторами) требуется дополнительный электронный компонент – транзистор. Транзистор – это полупроводниковый преобразовательный электроприбор. В схеме ниже транзистор усиливает ток и напряжение.

Есть биполярные транзисторы и униполярные (полевые). Подробнее про транзисторы читай здесь: <http://tiny.cc/a1-transistor>.

В схеме тебе понадобится защитный диод. Поясню, для чего он нужен. Электромотор может вырабатывать электричество, когда вращается без потребления электроэнергии, поэтому, если подключить к его выводам вольтметр и начать быстро вращать вал мотора вручную, на экране вольтметра ты увидишь изменение напряжения. Нет под рукой вольтметра – можно подсоединить маломощную лампочку, и она начнет тускло гореть, если быстро вращать вал мотора вручную.

Таким образом, электромотор может как потреблять электроэнергию, так и вырабатывать ее. На данном принципе построены электрогенераторы и системы рекуперации энергии, которые используются в машинах: электровозах, трамваях, троллейбусах, болидах «Форму-

лы 1» и простых электромобилях. В твоём случае, когда мотор тормозит, он вырабатывает ток, который пойдёт обратно на транзистор и повредит его. Для недопущения повреждения транзистора нужен диод, который способен пропускать ток только в одном направлении. Когда ток при свободном вращении мотора (торможение) пойдёт обратно, диод остановит его и не даст пойти обратно на транзистор.

Собирая эту схему, ты научишься подключать электромотор к Intel® Edison через транзистор и диод, а также управлять скоростью электромотора.

Что необходимо Дополнительные компоненты для сборки Схемы 13 «Электромотор, транзистор и диод». Приобрести их ты можешь по адресу: <http://tiny.cc/a1-motor-kit>.

Электромотор (DC 5 В).

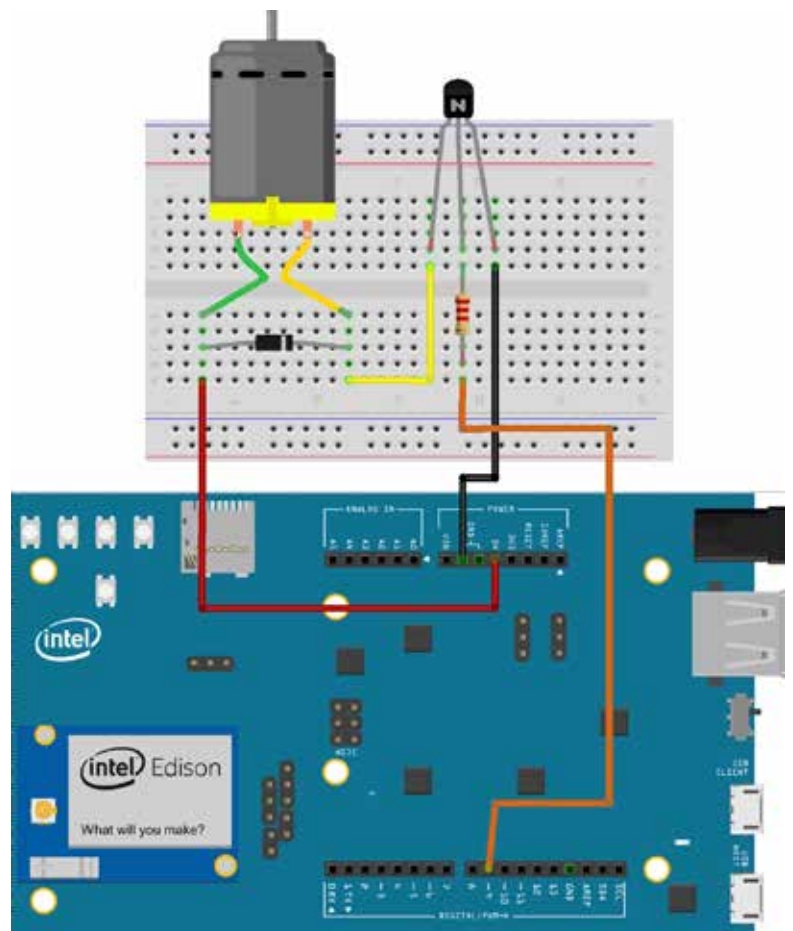
Транзистор (P2N2222A или аналог).

Диод (1N4001 или аналог).

Резистор 2.2 кОм.

Четыре провода «папа-папа».

Схема 13



Сборка схемы

- Действие 1.** Возьми электромотор и подсоедини его выводы на макетную плату. У электромотора есть два вывода, однако они не имеют полярности, поэтому неважно, какой вывод подключать к пину питания на плате 5 В, а какой к выводу транзистора. Если вручную поменять выводы мотора местами, вал электромотора начнет крутиться в другую сторону.
- Действие 2.** Установи диод на макетную плату. Обрати внимание на расположение светлой полоски на диоде. Вывод, который ближе к полоске, называется катодом, другой вывод – анодом. Ток может идти только от анода к катоду. Обратно он идти не может. Поэтому внимательно устанавливай диод, как показано на Схеме 13.
- Действие 3.** Теперь установи транзистор. Внимательно изучи его. Он имеет срезанную сторону, на которой видны цифры и буквы. По ним определяется модель транзистора. При установке его на макетную плату согласно Схеме 13 срезанная сторона с текстом должна смотреть в сторону платы. На Схеме 13 срезанная сторона с текстом отмечена буквой «N».
- Действие 4.** К центральному выводу транзистора (база) подключи один вывод резистора 2,2 кОм, второй вывод резистора подключи к цифровому пину на плате под номером 9.
- Действие 5.** Правый вывод транзистора (эмиттер) подключи к выводу заземления GND на плате.
- Действие 6.** Левый вывод транзистора (коллектор) подключи к выводу мотора так, как показано на Схеме 13.
- Действие 7.** Подключи второй вывод мотора к выводу питания 5 В на плате так, как показано на Схеме 13.
- Действие 8.** Запусти Arduino IDE и открой программу A1_A_plus_S13_Motor.ino.
- Действие 9.** Внимательно прочитай пояснения к программе и загрузи ее.

Результат. Вал мотора будет плавно разгоняться до максимальной скорости вращения, а потом так же плавно замедляться.

Не работает? Проверь соединения согласно Схеме 13.

Возможно, ты ошибся с подключением выводов транзистора или диода.

Если программа зависла, перезагрузи плату.

Попробуй догадаться сам, исходя из опыта, который ты приобрел, собирая предыдущие схемы.

СХЕМА 14 СЕРВОПРИВОД И УПРАВЛЕНИЕ ИМ

А вот теперь ты увидишь, где используется электромотор! Сервопривод – важная деталь во многих машинах. Он установлен в роботах-манипуляторах, станках с числовым и программным управлением, автомобилях и даже марсоходах. Сервопривод позволяет управлять скоростью движения и углом поворота выходного вала сервопривода согласно длительности ШИМ-импульса, который он принял. При этом сервопривод, в отличие от мотора непрерывного вращения, способен удерживать угол, на который был повернут. Например, чтобы манипулятор робота-марсохода повернуться на 90 градусов, на сервопривод подается команда об изменении угла поворота, после чего сервопривод выполняет команду и манипулятор поворачивается. Можно сказать, что сервопривод похож на человеческий сустав, который также способен вращаться и фиксировать свое положение. Однако, в отличие от сустава, вал сервопривода имеет только одну степень свободы, т. е. может вращаться вокруг одной оси. Про степени свободы хорошо написано тут: <http://tiny.cc/a1-servo>.



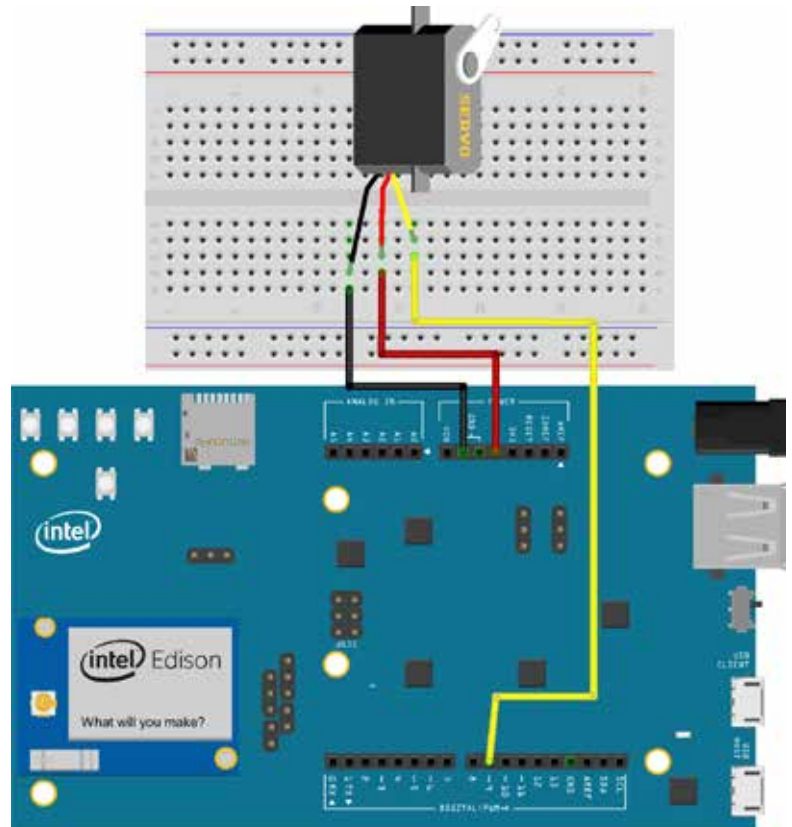
Внутри сервопривода находятся мотор непрерывного вращения, редуктор и специальная электроника (см. изображение выше). Все составные части сервопривода ты можешь видеть на рисунке. Благодаря электронике, сервопривод всегда знает, на какой угол повернут выходной вал (к нему крепится коромысло). Редуктор состоит из зубчатых колес разного диаметра и с разным количеством зубьев. Благодаря редуктору уменьшается количество оборотов электромотора и увеличивается крутящий момент на выходном валу редуктора. Подробнее про сервопривод можно прочесть тут: <http://tiny.cc/a1-servo2>.

Работая с этой схемой, ты освоишь основы управления сервоприводом и научишься использовать стандартную библиотеку Arduino Servo.

Что необходимо Сервопривод. Приобрести его ты можешь по адресу: <http://tiny.cc/a1-servo-kit>.

Три провода «папа-папа».

Схема 14



Сборка схемы

Действие 1. Рассмотрим сервопривод. К нему в комплекте идут различные коромысла (качалки), которые увеличивают рычаг на выходном валу сервопривода и тем самым увеличивают крутящий момент, с которым вал будет поворачиваться на заданный угол. Эта модель сервопривода имеет выходной крутящий момент на валу 1 кгс*см, или 0,1 Н*м. Подробнее про крутящий момент ты можешь почитать здесь: <http://tiny.cc/a1-servo-force>.

Сервопривод имеет три провода: черный или коричневый (GND, «земля»), красный (5 В), оранжевый или желтый (сигнальный провод). Данный сервопривод рассчитан на напряжение 5–6 В.

Действие 2. Соедини сервопривод с платой согласно Схеме 14.

Действие 3. Запусти Arduino IDE и открой программу A1_A_plus_S14_Servo.

Действие 4. Загрузи программу.

Результат

Выходное звено сервопривода вращается. Сначала вал поворачивается в одну сторону, потом обратно. Если поставить коромысло на сервопривод, ты сможешь понять, как изменяется сила, с которой вращается вал сервопривода.

Пояснения

Угол поворота сервомашинки зависит от ШИМ-сигнала, который был послан на сервопривод (подробнее см. [Главу 7](#). Широтно-импульсная модуляция).

В Arduino IDE для создания ШИМ-сигналов и управления сервоприводами и моторами существует специальная библиотека `Servo`. Она позволяет осуществлять программное управление сервоприводами через экземпляры класса `Servo`. Управление осуществляется следующими функциями:

`attach()` – прикрепляет экземпляр класса `Servo` к конкретному выводу на плате. Существуют два варианта синтаксиса для этой функции: `servo.attach(pin)` и `servo.attach(pin, min, max)`. При этом `pin` – номер вывода, к которому присоединяют сервопривод, `min` и `max` – длины импульсов в микросекундах, отвечающих за углы поворота 0° и 180° . По умолчанию выставляются равными 544 мкс и 2400 мкс соответственно.

`write()` – отдаёт команду сервоприводу принять некоторое значение параметра. Синтаксис следующий: `servo.write(angle)`, где `angle` – угол, на который должен повернуться сервопривод;

`writeMicroseconds()` – отдаёт команду послать на сервопривод импульс определённой длины, является низкоуровневым аналогом предыдущей команды. Синтаксис следующий: `servo.writeMicroseconds(uS)`, где `uS` – длина импульса в микросекундах;

`read()` – читает текущее значение угла, в котором находится сервопривод. Синтаксис следующий: `servo.read()`, возвращается целое значение от 0 до 180;

`detach()` – производит действие, обратное действию `attach()`, т. е. отсоединяет переменную от пина, к которому она была приписана. Синтаксис следующий: `servo.detach()`.

Не работает?

Проверь соединения согласно Схеме 14.

Проверь подключения выводов сервопривода к плате.

Если программа «зависла», перезагрузи плату.

Попробуй догадаться сам, исходя из опыта, который ты приобрел, собирая предыдущие схемы.

СХЕМА 14.1 ПОДКЛЮЧЕНИЕ СЕРВОПРИВОДА С ДОПОЛНИТЕЛЬНЫМ ИСТОЧНИКОМ ПИТАНИЯ

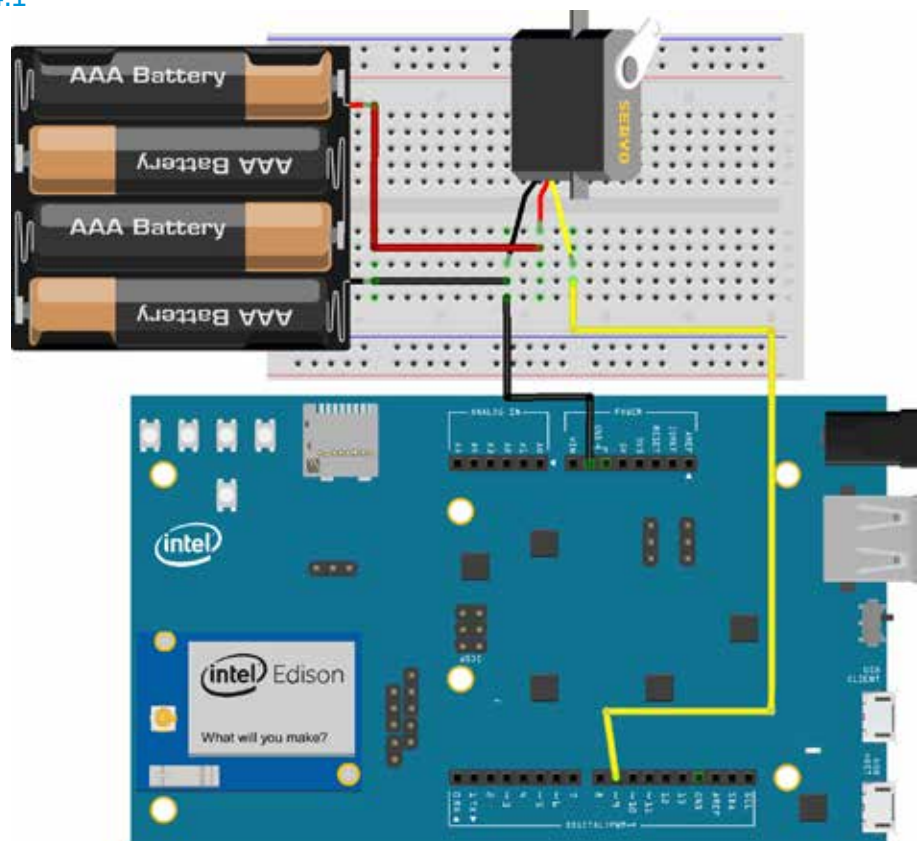
Так, как показано на Схеме 14, можно подключать только сервоприводы с низким энергопотреблением. Схема 14 создана исключительно в образовательных целях. Для подключения более мощных устройств нужен дополнительный источник питания, который будет питать силовую часть твоей схемы (сервопривод, мотор и т. д.), и источник питания, который будет питать управляющую часть твоей схемы (плата Intel® Edison).

Что необходимо Сервопривод.

Три провода «папа-папа».

Источник питания на 5 или 6 В. Идеально подойдет батарейный отсек для батареек типа AA или AAA. Подобные отсеки устанавливаются в игрушки.

Схема 14.1



Сборка схемы

- Действие 1.** Подключи черный провод (минус) от источника питания к черному проводу сервопривода (минус). Провода могут быть либо коричневые, либо черные. Это не принципиально.
- Действие 2.** Подключи красный провод (плюс) от источника питания с батарейками к красному центральному проводу сервопривода (плюс).

- Действие 3.** Сигнальный провод сервопривода подключи так же, как и в Схеме 14.
- Действие 4.** Подавай питание на плату через микро-USB или блок питания.
- Результат** Теперь твоя плата и сервопривод питаются от разных источников. Таким образом ты можешь создавать шагающих роботов, роботов-манипуляторов и т. д. Ведь теперь можно подключать более мощные сервоприводы. Однако батареек для них надолго не хватит. Потребуется Li-Po-аккумуляторы с напряжением 7,4 В или выше. Значение напряжения питания сервопривода надо уточнять в документации к устройству, т. к. при большом напряжении питания ты рискуешь повредить сервопривод, а при низком он не будет работать должным образом.

ГЛАВА 10

ОСВОЕНИЕ ОПЕРАЦИОННОЙ СИСТЕМЫ

YOCTO LINUX НА INTEL® EDISON

Многие вещи, описанные в этой главе, мы будем создавать напрямую в операционной системе (далее – ОС) Yocto Linux, под управлением которой работает Intel® Edison. В Intel® Edison стоит не полноценный Linux, а специальный дистрибутив для встраиваемых систем Yocto.

Подробнее о том, что представляет собой операционная система на ядре Linux, прочти здесь: <http://tiny.cc/a1-linux>.

Дополнительная информация и возможные правки к главе доступны по ссылке <http://tiny.cc/a1-edit-c10>.

Все стабильные исходники программ, которые были использованы, находятся по ссылке <http://tiny.cc/a1-tech-edison>. При возникновении трудностей попробуй работать с программами из этого источника.

ОС Linux многим отличается от Windows. Эта операционная система представляет собой полностью открытую UNIX-подобную бесплатную многопользовательскую ОС. Главное при работе в Linux – это уметь работать с командной строкой. Именно в командной строке пользователь вводит необходимые команды (многие из которых надо обязательно знать). ОС Linux заслуживает отдельной книги, поэтому рекомендую книгу по этой системе, а также скачать шпаргалку со всеми основными командами для командной строки в Linux. Все Linux-умения для работы с приведенными ниже схемами не понадобятся, однако, когда ты будешь иметь хотя бы общее представление о том, с чем работаешь, ты будешь лучше понимать, что делаешь.

Обратите внимание: что Intel® Edison не имеет графического интерфейса или видеовыхода. Поэтому управление микрокомпьютером происходит по последовательному порту с помощью клиента Putty. Все команды вводятся в окне клиента Putty, что для многих пользователей может показаться сложной или непривычной задачей.

Для того, чтобы подстегнуть интерес к изучению Linux, я скажу, что в основе ОС Android лежит ядро Linux и в ядре iOS, OS X тоже много схожего с ядром Linux. А теперь переходим к новым схемам!

10.1. Обновление операционной системы Intel® Edison

10.1.1. Вариант № 1 ОС Windows

Для начала, необходимо установить программу Putty, а также обновить ОС Intel® Edison до самой последней версии.

Если по каким-то причинам последняя версия прошивки для Intel® Edison будет давать сбой, по данной ссылке ты можешь скачать версию прошивки, на которой разрабатывались все схемы из книги: <http://tiny.cc/a1-edison-image>.

По ссылке необходимо выбрать File name: edison-image-ww18-15.zip и скачать zip-архив. После этого обновить Intel® Edison вручную согласно Инструкции № 4.

10.1.2. Установка Putty

Необходимо скачать Putty по ссылке <http://tiny.cc/a1-putty> для работы с Intel® Edison через COM-порт. Открой сайт. И кликни на ссылку, выделенную зеленым овалом.



После этого ты попадешь на сайт для загрузки приложения. Кликни на ссылку в красном овале, и начнется скачивание Putty. После скачивания открой .exe файл. Putty запустится.



10.1.3. Инструкция № 3 «Обновление Intel® Edison для Windows x64 (с помощью Intel® Software Setup Assistant)»

Шаг 1. Подключи Intel® Edison к компьютеру с помощью двух микро-USB-портов к USB-портам на компьютере.

Шаг 2. Теперь запусти программу Intel® Software Setup Assistant (установка программы в описана в Инструкции № 2), чтобы загрузить в микрокомпьютер последнюю прошивку.

Шаг 3. После запуска программы появится такое окно. Выбери Flash Image. Нажми на кнопку Next.



Шаг 4. В появившемся окне выбери Next . Начнется скачивание и загрузка последней прошивки, для этого потребуется некоторое время. Всего надо скачать около 200 Мб. К обновлению прошивки программа скачает и установит еще Intel® Phone Flash Tool Lite.



Шаг 5. Первым делом начнется скачивание программы Intel® Phone Flash Tool Lite.



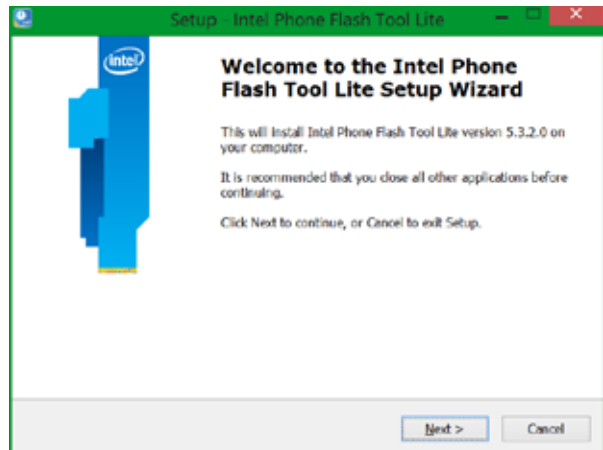
Потом – скачивание образа системы.



Шаг 6. После скачивания начнется установка программы Intel® Phone Flash Tool Lite.



Появится установщик программы. Здесь нет ничего сложного. Кликай во всех новых окнах Next, а в конце выбери Install. Ничего не меняй. В конце установки снимии галочку после Start Phone Flash Tool Lite after installation и кликай Finish.



Шаг 7. Появится такое окно. В нем тебе надо будет только отсоединить один из двух микро-USB-кабелей от платы. Отсоедини кабель от центрального микро-USB-порта (помечен цифрой 2) и быстро подсоедини обратно.



Шаг 8. Автоматически начнется обновление Intel® Edison. Этот процесс может занять до 20 минут и более.



Шаг 9. По окончании программа известит тебя об успешном обновлении Intel® Edison. Кликни Finish.



Если же по каким-то причинам у тебя не получилось обновить Intel® Edison с помощью данной программы, воспользуйся приведенной ниже Инструкцией № 4.

10.1.4. Инструкция № 4 «Обновление Intel® Edison для Windows x32 x64 (вручную)»

Обновлять ОС можно и вручную, без использования специальных программ. Это довольно просто. Инструкция подходит для любой разрядности ОС Windows.

Шаг 1. Для начала необходимо скачать последнюю версию ОС по ссылке <http://tiny.cc/a1-edison-downloads>, кликнув на странице ссылку с названием Release 2.1 Yocto* complete image (номер версии может отличаться). После этого начнется скачивание архива.

Шаг 2. Подключи Intel® Edison к компьютеру с помощью двух кабелей микро-USB к двум микро-USB-портам на плате Intel® Edison и двум USB-портам на компьютере соответственно.

Шаг 3. Теперь тебе надо открыть вкладку «Проводник» и узнать, под какой буквой находится диск Intel® Edison. Буквы могут быть разные, у меня это диск E : .

Ниже я буду описывать все для моего случая, поэтому если у тебя другая буква, то просто замени на нее везде букву моего диска.



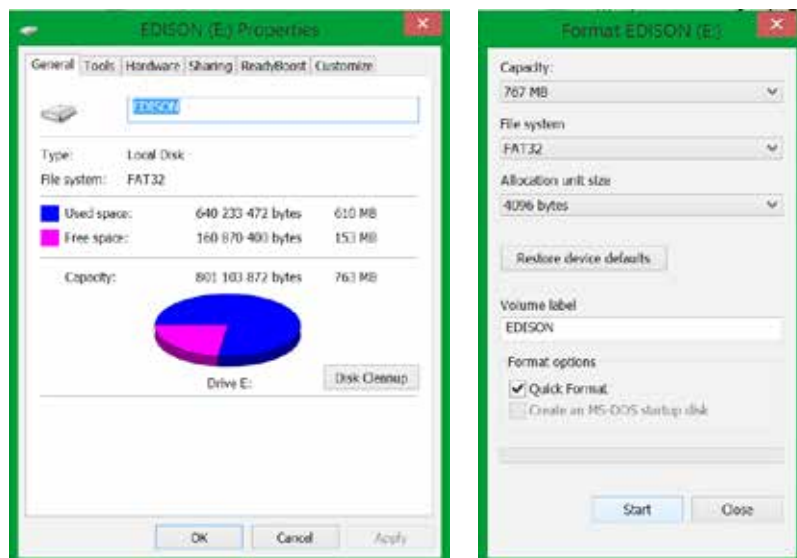
10.1.5. Форматирование встроенного накопителя Intel® Edison из FAT в FAT32

Шаг 4. Тебе надо форматировать диск Edison (E :) в формат FAT32, т. к. по умолчанию Intel® Edison из коробки идет с файловой системой FAT. Чтобы узнать тип файловой системы, открой меню «Свойства», кликнув правой кнопкой мыши по диску и выбрав «Свойства». Если у тебя уже стоит FAT32, пропусти Шаг 4.

Форматирование является обязательным действием. Если ты когда-нибудь форматировал флешку, то справишься и с этим заданием.

Клики правой кнопкой мыши по диску, выбери из списка «Форматировать» и выстави File system FAT32; название диска можешь взять любое, я оставил по умолчанию: Edison. Прежде чем продолжать, внимательно посмотри, верный ли это диск Intel® Edison (буква диска). Случайно ты можешь удалить всю информацию с другого нужного тебе диска!

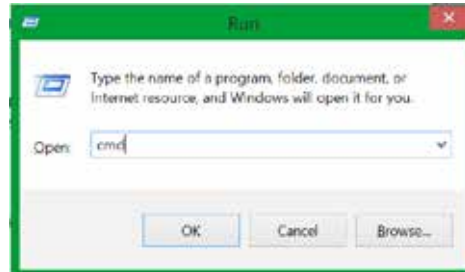
Теперь запускай форматирование диска.



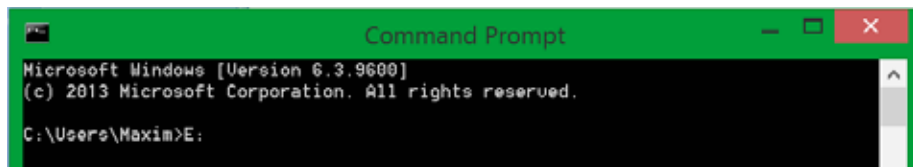
Шаги 5-9 из пункта 10.1.5 выполнять не надо, если ты только что выполнил Шаг 4 из пункта 10.1.5 и отформатировал внутренний накопитель Intel® Edison. Однако если ты хочешь поменять установленную тобой ОС на более новую, то Шаги 5–9 обязательны к выполнению.

Продолжение инструкции № 4

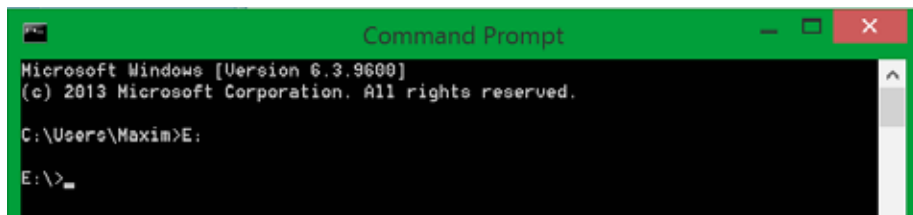
Шаг 5. Необходимо открыть командную строку. Это можно сделать, набрав в поиске Windows слово «Выполнить» (если английская версия Windows, то Run), открыв появившуюся иконку и набрав cmd. Потом надо нажать Ок. (Также открыть командную строку можно любым другим известным тебе способом.)



Шаг 6. Теперь снова откроем командную строку. В ней наберем E:

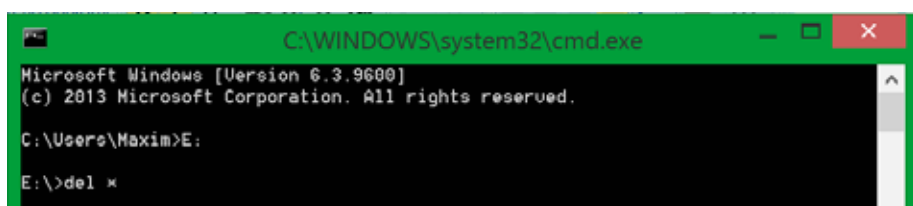


Шаг 7. Жмем клавишу Enter. Получится следующее.



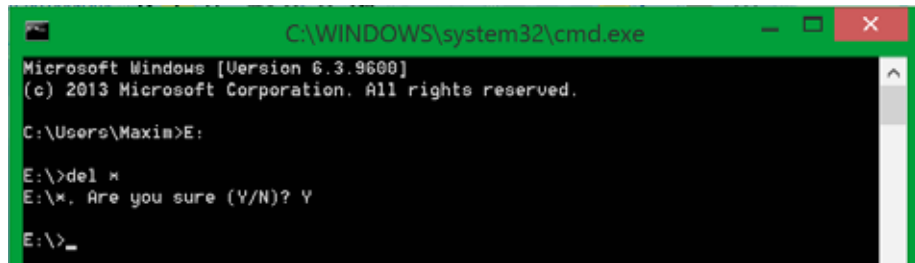
Шаг 8. Теперь удали всю информацию с диска, чтобы потом скопировать на него новую прошивку. Прежде чем продолжать, внимательно посмотри, верный ли это диск Intel® Edison (буква диска). Случайно ты можешь удалить всю информацию с другого нужного тебе диска!

Вводим `del *` (слово `del` с пробелом, после которого идет символ «звездочка») и нажимаем клавишу `Enter`. Если у тебя полностью новая плата из коробки, то, скорее всего, ты ничего не сможешь удалить данной командой, и после Шага 8 появится надпись: «Невозможно найти диск». Тогда переходи к Шагу 10. Однако если ты меняешь старую прошивку на новую, то Шаги 7, 8, 9 должны быть выполнены.



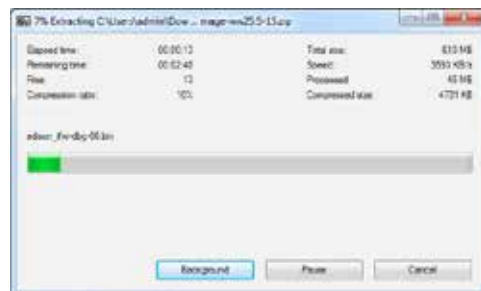
Шаг 9. В вопросе Are you sure (Y/N) выбираем Y и опять жмем Enter, тем самым удаляя все с накопителя микрокомпьютера Intel® Edison.

Если в ходе этих действий возникли вопросы, то пиши maxliferobot@gmail.com с темой сообщения «Обновление ОС Intel® Edison вручную».

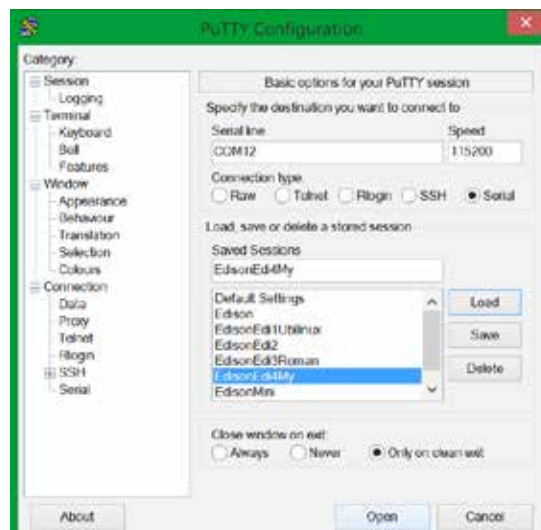


Шаг 10. Установи архиватор 7-Zip <http://tiny.cc/a1-7zip>.

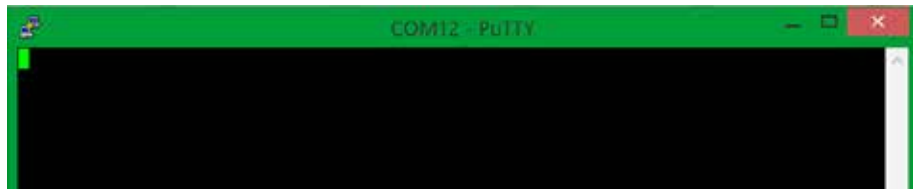
После того как установишь архиватор, извлеки скачанный архив с операционной системой Intel® Edison (название архива должно быть приблизительно таким: edison-image-ww18-15, цифры могут различаться) на диск накопителя Intel® Edison. В моем случае это был диск E. В настройках архиватора поставь «Перезаписывание файлов без уведомления». Подожди минут пять. После того как архив разархивируется, переходи к следующему шагу.



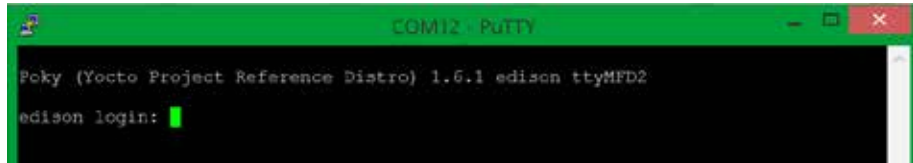
Шаг 11. Открой Putty, если еще не открыл. Выбери Serial. В текстовом окне слева напиши на английском COM и номер USB Serial Port, у меня COM12. (Как узнать номер USB Serial Port, сказано в Шаге 14, Инструкции № 2). Выставляем скорость 115200. Выбираем Open.



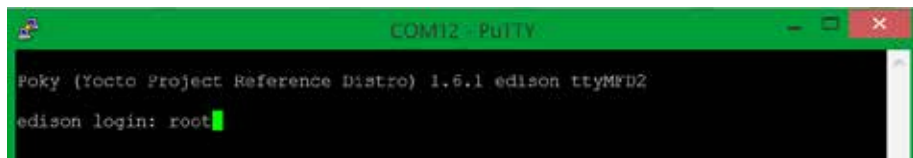
Шаг 12. Появится пустое окно консоли.



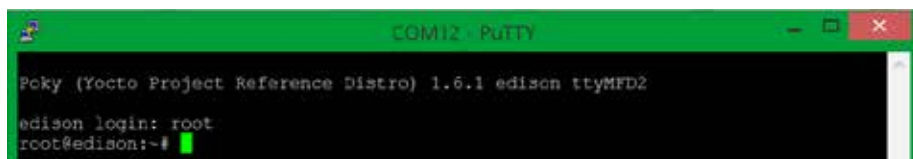
Шаг 13. После того как появится окно, ждем на клавишу `Enter` на клавиатуре два раза. (В некоторых случаях достаточно и одного раза.) Потом должно быть так:



Шаг 14. Ставим курсор и пишем `root`. Нажимаем `Enter`. Тем самым мы зашли в ОС с правами суперпользователя (`root`).



Шаг 15. Когда ты зашел в ОС, окно должно выглядеть так:



Шаг 16. Введи команду `reboot ota` и нажми на клавишу `Enter`. Начнется обновление ОС. Жди около пяти минут, пока не перестанет меняться текст в окне и не появится `edison login:` как в Шаге 13.

Результат. Твоя операционная система Yocto Linux обновлена.

Продолжаем создание твоего устройства на базе Intel® Edison.

10.1.6. Вариант № 2 ОС Mac OS X

Шаг 1. Подключи Intel® Edison к компьютеру с помощью двух проводов из набора к двум микро-USB-портам на плате Intel® Edison и двум USB-портам на компьютере соответственно. Выполни Шаг 4 Инструкция № 4 «Обновление Intel® Edison для Windows x32 x64 (вручную)». В Шаге 4 тебе надо только отформатировать внутренний накопитель Intel® Edison, некоторые действия могут различаться.

Шаг 2. Выполни все действия, указанные на странице по этой ссылке: <http://tiny.cc/a1-edison-os-x-terminal>.

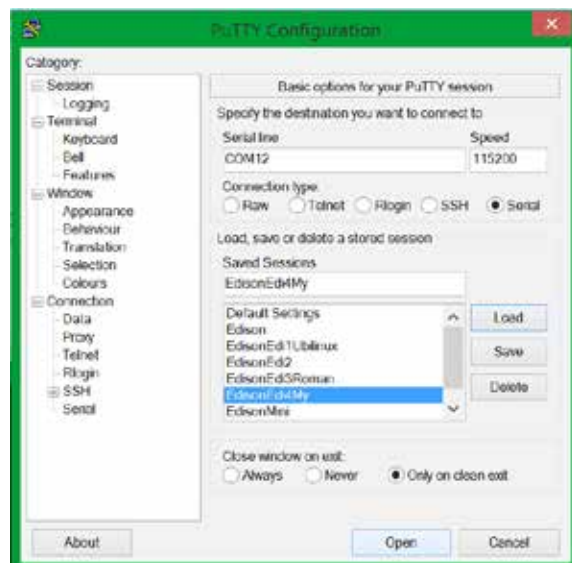
Шаг 3. Обнови Intel® Edison, как указано в ссылке <http://tiny.cc/a1-edison-os-x-update>.

Шаг 4. Теперь ты можешь выполнять все действия и собирать схемы далее по книге. Учти только, что вместо Putty нужно будет открывать Terminal–приложение.

10.2. Подключение Intel® Edison к Wi-Fi

Чтобы работать с облаком, нам, естественно, требуется подключение к интернету. Подключим Intel® Edison к Wi-Fi-точке доступа, которая имеет выход в интернет.

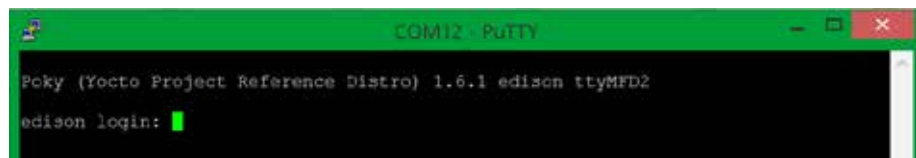
Шаг 1. Открой Putty. Выбери `Serial`. В текстовом окне слева напиши на английском COM и номер USB Serial Port, у меня COM12. Как узнать номер USB Serial Port, сказано в, Инструкции № 2. Выставляем скорость 115200. Выбираем `Open`.



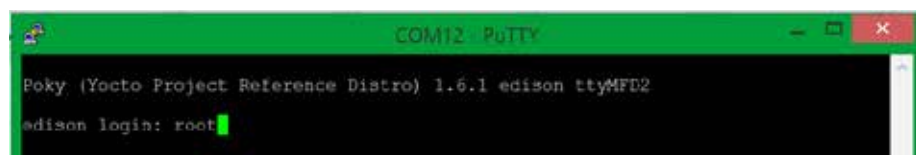
Шаг 2. Появляется пустое окно.



Шаг 3. Когда появится пустое окно, нажми клавишу `Enter` на клавиатуре два раза. (В некоторых случаях достаточно и одного.) Потом должно быть так.



Шаг 4. Поставь курсор и напиши `root`. Потом `Enter`. Тем самым ты зайдешь в операционную систему микрокомпьютера Intel® Edison с правами суперпользователя (`root`).



Шаг 5. Когда ты введешь логин и нажмешь Enter, окно должно выглядеть так:

```
COM12 - PuTTY
Poky (Yocto Project Reference Distro) 1.6.1 edison ttyMFD2
edison login: root
root@edison:~#
```

Шаг 6. Введи команду `configure_edison --w` и нажми Enter.

```
edison login: root
root@edison:~# configure_edison --w
```

Начнется сканирование сетей Wi-Fi.

```
COM12 - PuTTY
Configure Edison: WiFi Connection
Starting scan
```

Шаг 7. Введи номер, под которым находится твоя сеть Wi-Fi. У меня это номер 4. (Моя сеть называется maksim.) Нажми Enter.

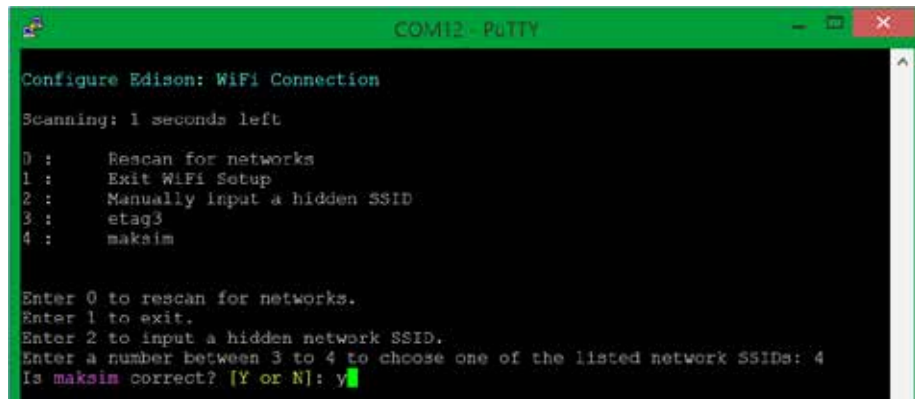
```
COM12 - PuTTY
Configure Edison: WiFi Connection
Scanning: 1 seconds left
0 : Rescan for networks
1 : Exit WiFi Setup
2 : Manually input a hidden SSID
3 : etag3
4 : maksim

Enter 0 to rescan for networks.
Enter 1 to exit.
Enter 2 to input a hidden network SSID.
Enter a number between 3 to 4 to choose one of the listed network SSIDs:
```

Шаг 8. Тебя попросят подтвердить название сети. Если это твоя сеть, введи «Y» и жми Enter.

```
COM12 - PuTTY
Configure Edison: WiFi Connection
Scanning: 1 seconds left
0 : Rescan for networks
1 : Exit WiFi Setup
2 : Manually input a hidden SSID
3 : etag3
4 : maksim

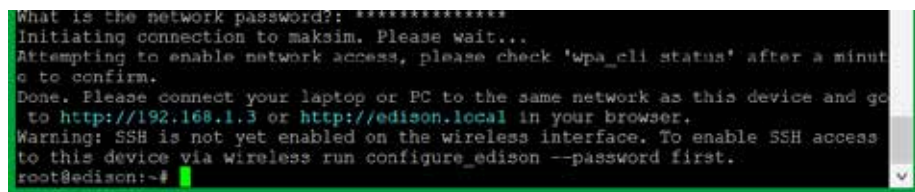
Enter 0 to rescan for networks.
Enter 1 to exit.
Enter 2 to input a hidden network SSID.
Enter a number between 3 to 4 to choose one of the listed network SSIDs: 4
Is maksim correct? [Y or N]:
```

Шаг 9. Тебе предложат ввести пароль. Сделай это. Если ошибся, то просто нажми сочетание клавиш Ctrl+C и начни вводить команды заново.

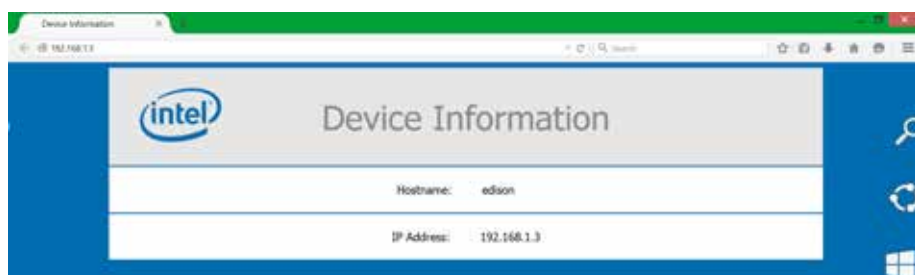


Результат. Если ты все сделал правильно, у тебя появится следующий текст: «192.168.1.3. является IP-адресом платы». Скорее всего, у тебя будет другой IP-адрес.



Проверка. Открой браузер на своем компьютере (он должен быть подключен к той же сети, что и Intel® Edison). Потом набери в адресной строке IP-адрес платы (будет выделен голубым) и нажми Enter.

Загрузится страница платы Intel® Edison. Мы в сети!



ГЛАВА 11

ИНТЕРНЕТ ВЕЩЕЙ И INTEL® EDISON.

РАБОТА С ОБЛАКОМ INTEL® IOT

DASHBOARD

11.1. Введение в IoT (Internet of Things) и облачный сервис Intel® IoT Analytics

Дополнительная информация и возможные правки к главе доступны по ссылке <http://tiny.cc/a1-edit-c1>.

Intel® Edison изначально поддерживает языки разработки C, C++, Python и Node.js/Javascript для создания IoT-приложений. Поддерживает большое количество IDE: Arduino, Eclipse, Intel® XDK, Wylidrin.

Более продвинутым пользователям рекомендую изучить раздел Intel® Developer Zone по ссылке: <http://tiny.cc/a1-edison-hardware>. Там подробно описано все, что можно реализовать на плате Intel® Edison. Также существуют и облачные сервисы от Google Cloud, Microsoft Azure, IBM IoT Foundation <http://tiny.cc/a1-iot-clouds>.

Для тебя сейчас главное – получить общее представление об облачных сервисах, поэтому далее мы будем работать с наиболее дружелюбным сервисом Intel® IoT Analytics и задействуем уже известную нам Arduino IDE, но только на более высоком уровне программирования.

Начинающему изобретателю интересно собирать простые схемы, но сам микрокомпьютер Intel® Edison создавался прежде всего для интеграции с носимыми и миниатюрными устройствами, которые бы работали с интернетом вещей.

Сегодня направление носимых и «умных» устройств стремительно развивается и я предлагаю прикоснуться к нему. Каждый день многие привычные вещи становятся «умнее»: современные холодильники уже умеют выходить в интернет, наручные часы позволяют играть, выходить в интернет и т. д. Так почему бы не объединить их в одно целое, подумали в компании Intel®.

Сегодня уже недостаточно иметь устройство с датчиками, которые локально собирают данные, ведь можно подключить миллионы устройств к Сети и получать данные на единый облачный сервис с целью проведения мониторинга и оценки поступающей информации. Таким же образом устройство может получать из Сети команды. Это удобно и революционно. Ожидается, что к концу 2020 года количество устройств, подключенных к интернету вещей, составит 50 миллиардов! И все эти устройства будут общаться между собой, взаимодействовать, формируя единую, целостную систему – социальную сеть для электроники.

Подробнее про IoT (интернет вещей) читай здесь <http://tiny.cc/a1-iot>.

Для реализации этой задачи корпорация Intel® создала специальный облачный сервис под названием Intel® IoT Analytics. Главной его целью является объединение всех IoT-устройств на одном портале и обеспечение удобного взаимодействия. Зайти на этот портал ты можешь по ссылке <http://tiny.cc/a1-iot-analytics>.

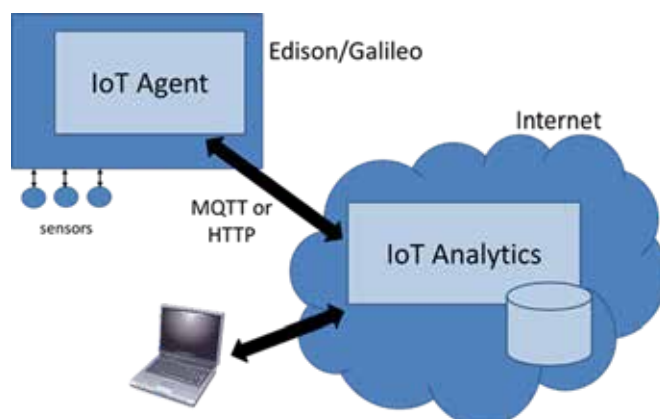
Чуть позже ты создашь там учетную запись пользователя и начнешь делать свой проект.

Возможности Intel® IoT Analytics следующие:

- Объединение всех устройств на одном портале управления.
- Отправка данных с датчиков устройств.
- Управление устройствами из любой точки мира.
- Построение графических изображений, фиксирующих, как меняются полученные с устройств данные.
- Создание правил работы для подключенных устройств.

Это не все возможности, но на первом этапе работы их будет достаточно.

Взаимодействие устройства Intel® Edison и облака Intel® IoT Analytics



Как видно на рисунке, команды с управляющего компьютера подаются на облачный сервис, а оттуда – к устройству (IoT Agent) по протоколу MQTT. Все соединения двухсторонние. Твой управляющий компьютер может как отправлять информацию на устройство, так и получать ее с устройства. То же самое касается устройства (IoT Agent).

Кстати, в роли управляющего компьютера может выступать такое же IoT-устройство, которое будет управлять другим девайсом без участия человека. Вот что значит интернет вещей! Это своего рода «социальная» сеть для всех компьютеров, смартфонов, бытовых приборов и других гаджетов. Такая сеть позволяет устройствам общаться между собой и выполнять свои функции еще качественнее и быстрее!

Компоненты сервиса Intel® IoT Analytics

Для обмена данными с устройством используются компоненты двух типов:

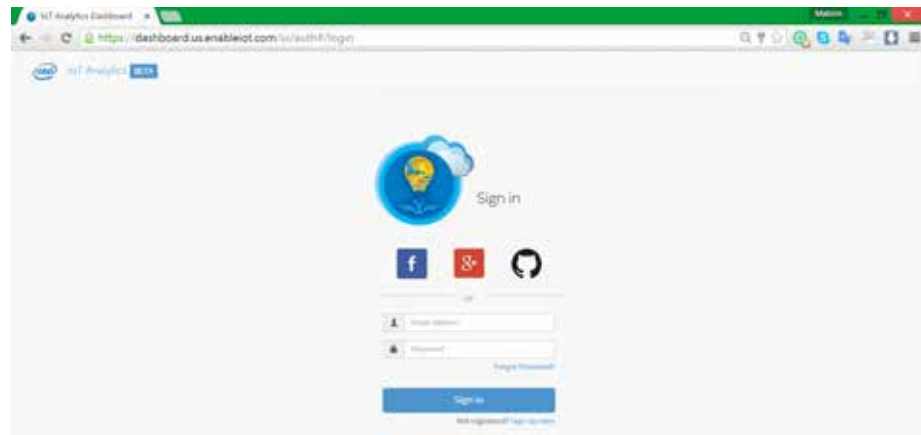
- **Sensor** (датчик, сенсор). Этот компонент позволяет получать данные от датчиков, которые подключены к IoT-устройству. Это могут быть влажность, температура воздуха и т. д. Информация с датчиков поступает не мгновенно, а с некоторой задержкой, что обусловлено самой архитектурой облачного сервиса и скоростью передачи данных на большие расстояния.
- **Actuator** (актуатор или регулятор). Такой компонент позволяет удаленно управлять устройством. Выключить или включить светодиод, привести в движение сервопривод и все то, что подключено к устройству. К сожалению, большая задержка (около 5 секунд), после отправки команды на устройство, не позволяет использовать компонент Actuator для управления в режиме реального времени. Если ты захочешь построить робомобиль, то завести его удаленно ты сможешь, а вот управлять им, скорее всего, нет. Тебе придется просчитывать его траекторию, потому что в случае неудачи команда «стоп» придет только через 5 секунд. Однако есть и плюсы: ты сможешь сделать свой марсоход и смоделировать управление им (к примеру, для школьного или студенческого проекта), т. к. настоящие команды доходят до Марса приблизительно за 20 минут.

Перед тем как создать Sensor или Actuator, тебе нужно указать «Тип компонента». Он включает в себя тип передаваемых данных (числа или символы), пороги изменения значений (минимальное и максимальное), единицу измерения величины (кг, см, Н и т. д.) Для простоты работы на портале Intel® IoT Analytics указаны стандартные типы компонентов: датчик температуры, датчик влажности, переключатель. Каждый тип имеет свою версию. Когда ты меняешь свойство типа компонента, версия меняется. Все датчики привязаны к устройствам, а устройства – к административной записи на портале.

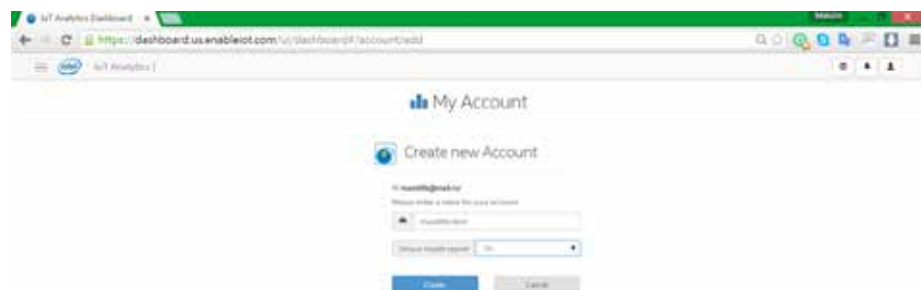
В этой части мы рассмотрим несколько электрических схем, а также изучим облачный сервис Intel® IoT Dashboard, чтобы ты получил общее представление по работе с облаком и смог увидеть безграничные возможности Intel® Edison.

11.2. Создание учетной записи на портале Intel® IoT Analytics

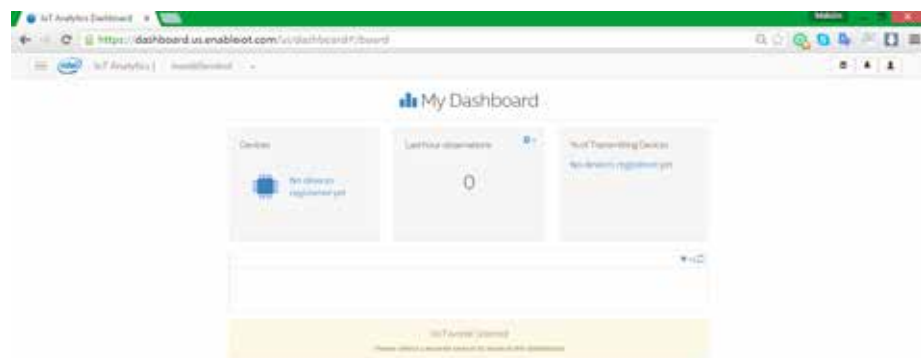
Шаг 1. Зайди на <http://tiny.cc/a1-iot-analytics-login> и зарегистрируйся. Можно зайти при помощи аккаунта Facebook, Google+ Github.



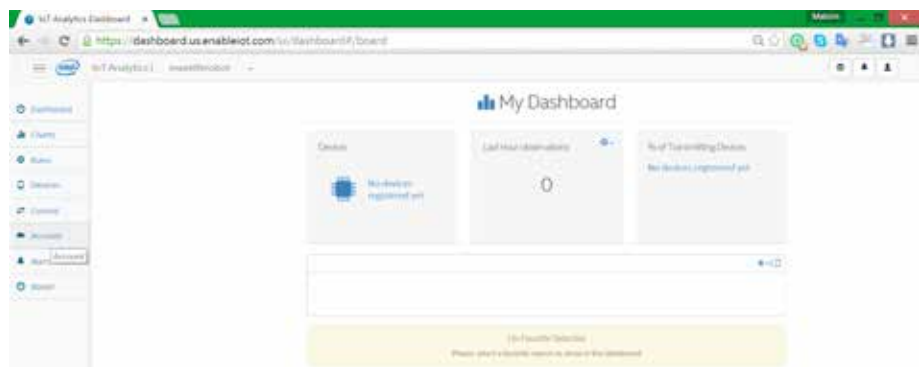
Шаг 2. Теперь тебе надо указать название своего облака. Мое называется `maxxliferobot`. В поле `Sensor health report` поставь `On`. Нажми `Create`.



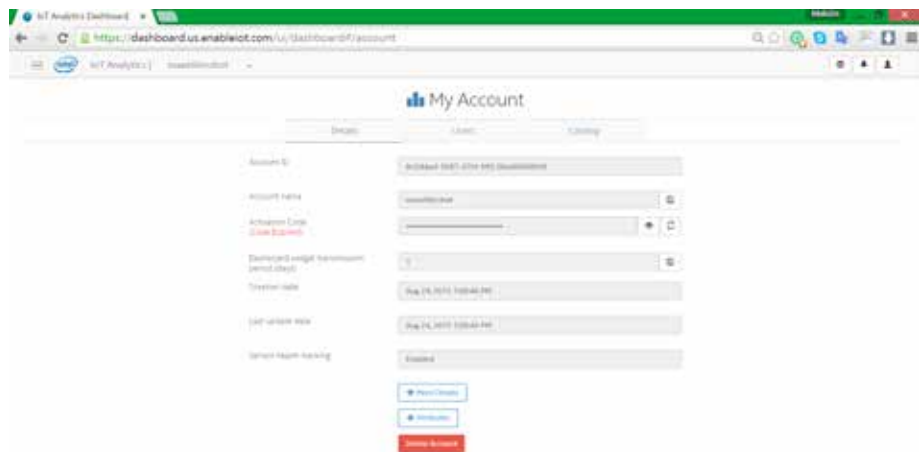
Шаг 3. После указания имени появится панель управления. Пока 0 зарегистрированных устройств.



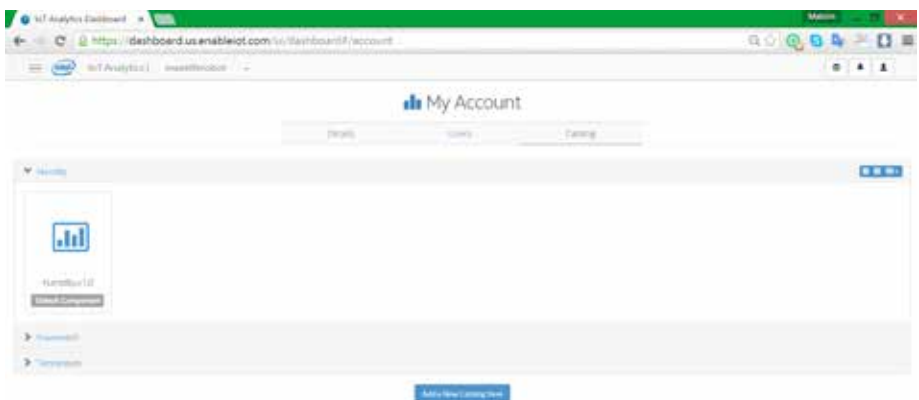
Шаг 4. Теперь требуется выполнить регистрацию типов датчиков на портале. Кликни на кнопку в верхнем левом углу, рядом с логотипом Intel®. Появится панель управления.



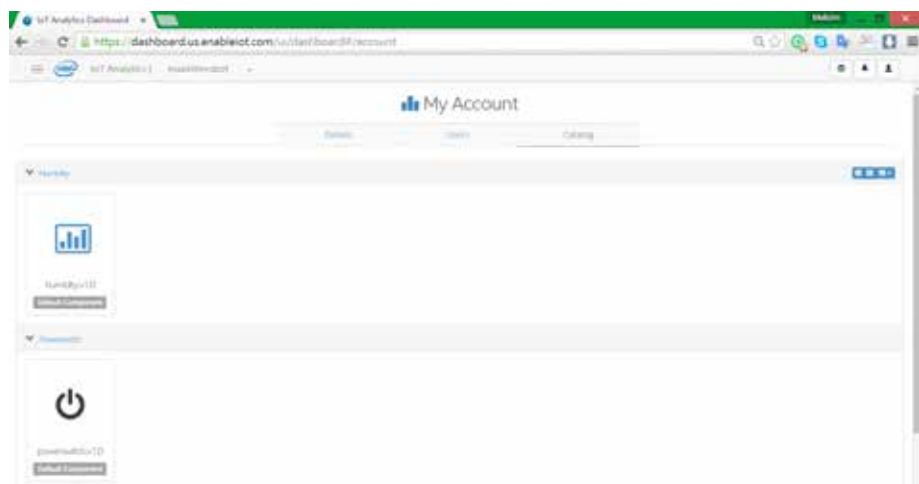
Шаг 5. Перейди в раздел Account.



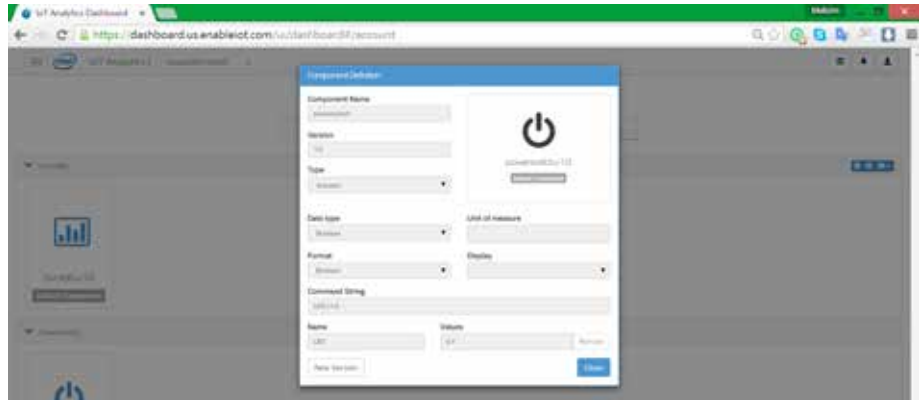
Шаг 6. Выбери вкладку Catalog. Здесь уже зарегистрировано несколько типов датчиков.



Шаг 7. Открой вкладку Powerswitch. Это выключатель светодиода.



Шаг 8. Открой компонент Powerswitch. Появится окно, в котором можно увидеть все свойства тестового компонента, а при помощи кнопки `New Version` – создать новый компонент класса со своими свойствами.



На этом работу с облачным сервисом на время откладываем в сторону. И переходим к настройке платы Intel® Edison.

11.3. Настройка Intel® Edison и регистрация на портале Intel® IoT Analytics

Подразумевается, что у тебя открыта Putty и ты вошел в систему Intel® Edison с правами пользователя `root`. Плата должна быть подключена к Wi-Fi ([Глава 10.2](#)). После ввода команд, чтобы они были выполнены, всегда необходимо нажимать клавишу `Enter`.

Шаг 1. Выполни тест для агента `iotkit`. Введи команду `iotkit-admin test`

Должно быть примерно как на картинке. Если все хорошо, значит, тест подключения к облаку пройден успешно.

```
root@edison:~# iotkit-admin test
2015-08-24T15:56:22.754Z - info: Trying to connect to host ...
2015-08-24T15:56:24.494Z - info: Connected to dashboard.us.enableiot.com
2015-08-24T15:56:24.498Z - info: Environment: prod
2015-08-24T15:56:24.500Z - info: Build: 0.14.2
root@edison:~#
```

Шаг 2. Тебе надо узнать ID твоей платы Intel® Edison. Введи команду `iotkit-admin device-id`

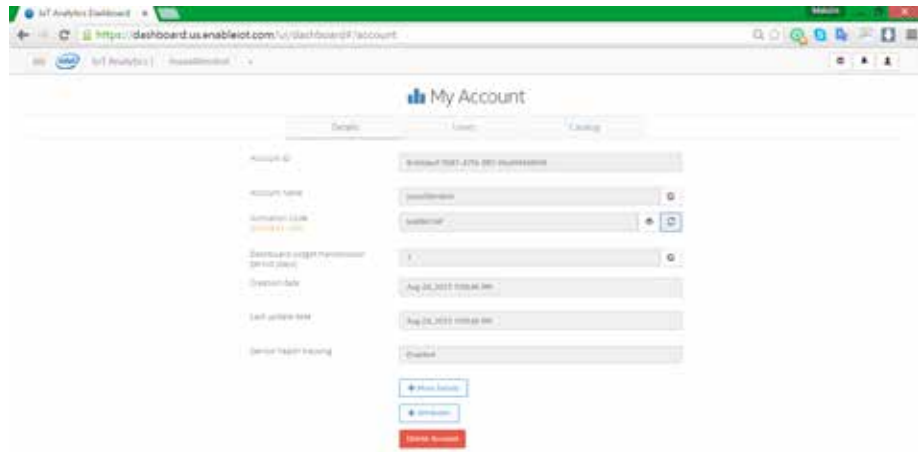
```
root@edison:~# iotkit-admin test
2015-08-24T15:56:22.754Z - info: Trying to connect to host ...
2015-08-24T15:56:24.494Z - info: Connected to dashboard.us.enableiot.com
2015-08-24T15:56:24.498Z - info: Environment: prod
2015-08-24T15:56:24.500Z - info: Build: 0.14.2
root@edison:~# iotkit-admin device-id
2015-08-24T15:58:44.529Z - info: Device ID: aa-30-e4-59-90-eb
root@edison:~#
```

Шаг 3. Теперь необходимо зарегистрировать устройство в облаке. Для этого перейдем на портал `dashboard.us.enableiot.com`, раздел `Account`, вкладка `Details`.

Получи код из строки `Activation Code`.

Чтобы это сделать, сгенерируй новый код, нажав на крайнюю правую кнопку со стрелками; чтобы увидеть код, нажми на кнопку с пиктограммой «глаза».

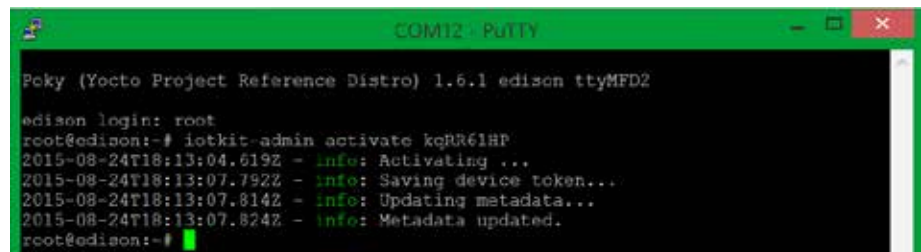
Код нельзя скопировать, поэтому запомни его или запиши на бумажке. Далее он понадобится.



Шаг 4. Теперь вернись к консольному окну и введи команду `iotkit-admin activate kod1234`

Вместо `kod1234` вводи свой сгенерированный уникальный код.

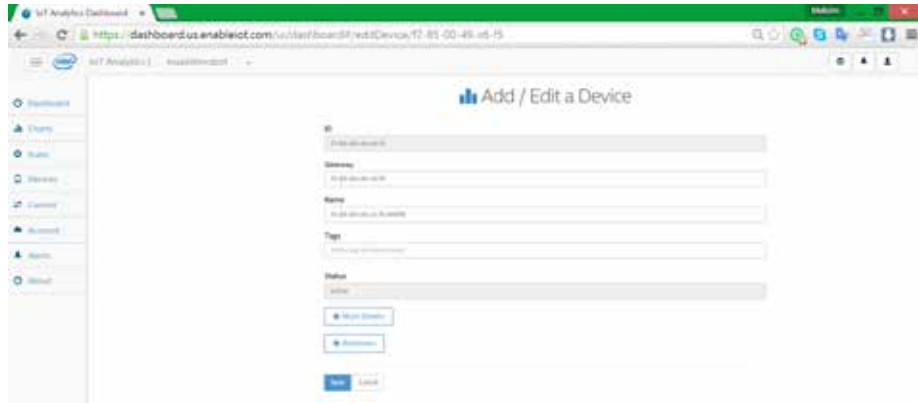
Если ввел данные верно, текст будет светиться зеленым.



Шаг 5. Возвращайся на портал IoT Dashboard. Теперь на вкладке Devices появилось первое зарегистрированное устройство.



Шаг 6. Выбери устройство – откроется страница редактирования параметров. В поле Status значение изменилось на active.



11.4. Установка библиотек IoTKit для Arduino IDE

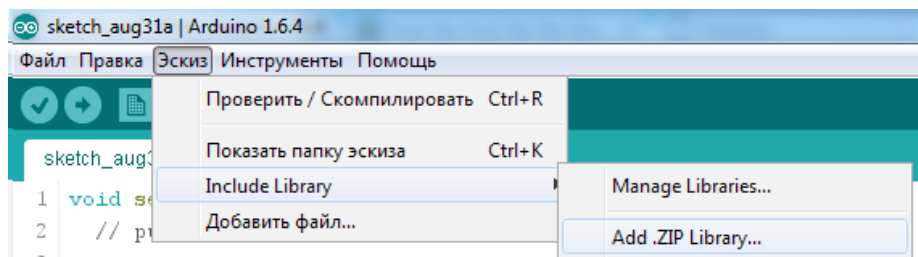
Для создания схем уровня В тебе необходимо установить библиотеки, чтобы успешно скомпилировать новые программы.

Шаг 1. Скачай архив по ссылке <http://tiny.cc/a1-iot-library> или https://github.com/mrobotby/A1_lot_kit_library.

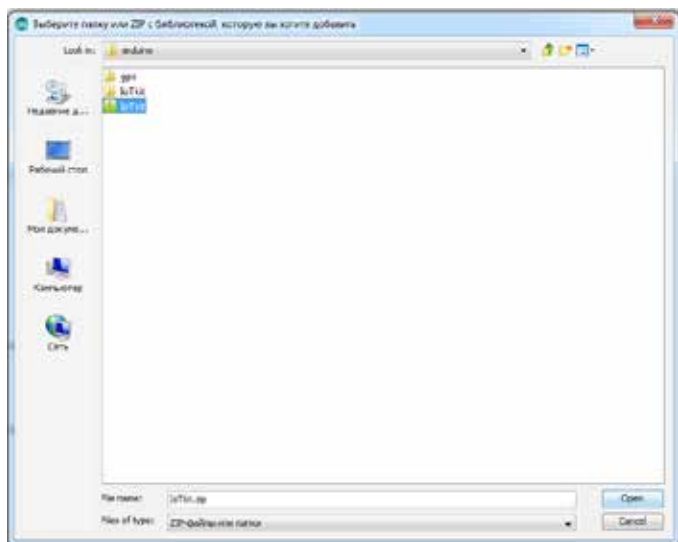
Справа будет кнопка Download ZIP. Скачай архив.

Шаг 2. Распакуй архив в одноименную папку.

Шаг 3. Открой Arduino IDE. Выбери вкладку «Эскиз», Include Library, Add .ZIP Library.



Шаг 4. Укажи путь к твоему распакованному архиву. Ищи в папке A1_lot_kit_library-master папку A1_lot_kit_library-master, а в ней ZIP-архив IoTkit. Укажи путь именно к архиву. Нажми Open. Библиотека добавлена.



Схемы уровня сложности В

СХЕМА В1

УПРАВЛЕНИЕ СВЕТОДИОДОМ С ПОМОЩЬЮ ОБЛАКА INTEL® IOT DASHBOARD

Мы будем зажигать светодиод на плате Intel® Edison, управляя им из интернета. Таким образом с помощью Intel® Edison можно сделать «умный» дом или еще какое-нибудь устройство, для которого необходимо удаленное управление. Т. к. плата имеет Wi-Fi, а на борту микрокомпьютера ОС Yocto Linux, то мы можем задействовать стандартные функции в микрокомпьютере и наладить взаимодействие со скетчем Arduino, чтобы упростить создание прототипов.

Что необходимо Плата Intel® Edison.

Два микро-USB-кабеля.

Подразумевается, что у тебя открыта Putty и ты вошел в систему Intel® Edison с правами пользователя root, а также, что плата подключена к Wi-Fi.

Настройка и сборка схемы

Действие 1. Зайди в консоль и выполни регистрацию виртуального переключателя.

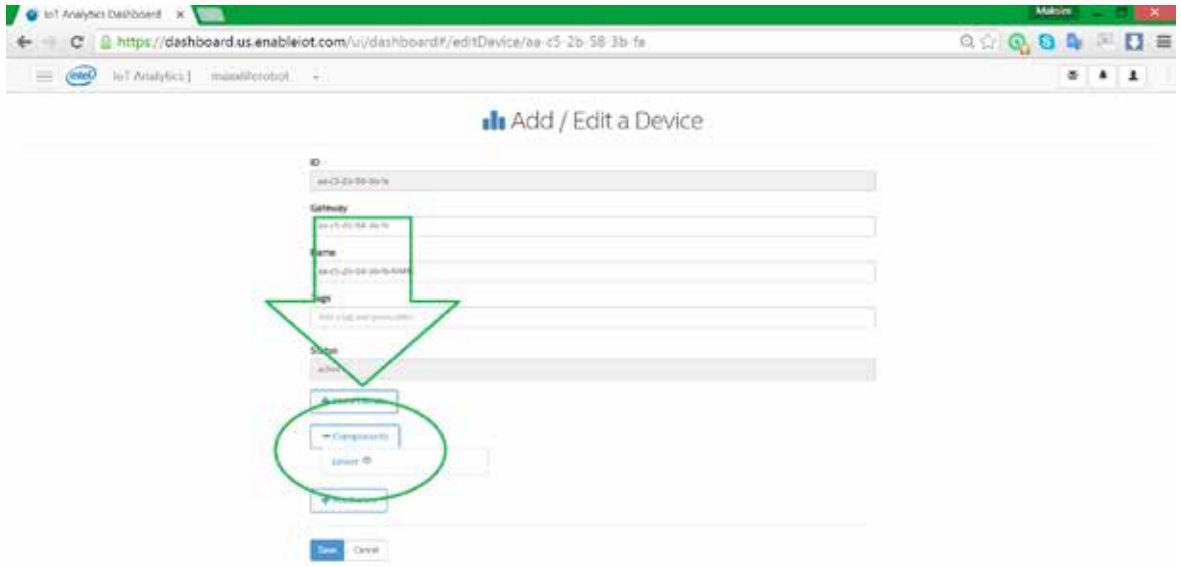
Команда для регистрации: `iotkit-admin register имя_датчика тип_датчика`

Введи команду для своего случая: `iotkit-admin register power powerswitch.v1.0`

```
COM12 - PuTTY
Poky (Yocto Project Reference Distro) 1.6.1 edison ttyMFD2
edison login: root
root@edison:~# iotkit-admin activate kqRR61HP
2015-08-24T18:13:04.619Z - info: Activating ...
2015-08-24T18:13:07.792Z - info: Saving device token...
2015-08-24T18:13:07.814Z - info: Updating metadata...
2015-08-24T18:13:07.824Z - info: Metadata updated.
root@edison:~# iotkit-admin register power powerswitch.v1.0
```

```
COM12 - PuTTY
Poky (Yocto Project Reference Distro) 1.6.1 edison ttyMFD2
edison login: root
root@edison:~# iotkit-admin activate kqRR61HP
2015-08-24T18:13:04.619Z - info: Activating ...
2015-08-24T18:13:07.792Z - info: Saving device token...
2015-08-24T18:13:07.814Z - info: Updating metadata...
2015-08-24T18:13:07.824Z - info: Metadata updated.
root@edison:~# iotkit-admin register power powerswitch.v1.0
2015-08-24T18:23:17.259Z - info: Starting registration ...
2015-08-24T18:23:17.297Z - info: Device has already been activated. Updating ...
2015-08-24T18:23:17.305Z - info: Updating metadata...
2015-08-24T18:23:17.353Z - info: Metadata updated.
Attributes sent
2015-08-24T18:23:23.307Z - info: Component registered name=power, type=powerswitch.v1.0, cid=f862c0f3-8538-4cad-8792-39e501cf198b, d_id=f2-85-00-49-c6-f5
root@edison:~#
```

После регистрации компонента в разделе Add/Edit a Device в пункте Components должен появиться объект power.



Действие 2. Запусти агента на своем микрокомпьютере командой `systemctl start iotkit-agent`

После ввода нажми Enter.

После каждой перезагрузки микрокомпьютера Intel® Edison, необходимо выполнять Действие 2, т. е. запускать агента на плате.

Дополнительные команды для управления агентом:

`systemctl stop iotkit-agent` – остановка агента;

`systemctl restart iotkit-agent` – перезапуск агента;

`systemctl status iotkit-agent` – узнать текущее состояние агента;

```
COM12 - PuTTY

Poky (Yocto Project Reference Distro) 1.6.1 edison ttyMFD2

edison login: root
root@edison:~# iotkit-admin activate kgRR61HP
2015-08-24T18:13:04.619Z - info: Activating ...
2015-08-24T18:13:07.792Z - info: Saving device token...
2015-08-24T18:13:07.814Z - info: Updating metadata...
2015-08-24T18:13:07.824Z - info: Metadata updated.
root@edison:~# iotkit-admin register power powerswitch.v1.0
2015-08-24T18:23:17.259Z - info: Starting registration ...
2015-08-24T18:23:17.297Z - info: Device has already been activated. Updating ...
2015-08-24T18:23:17.305Z - info: Updating metadata...
2015-08-24T18:23:17.353Z - info: Metadata updated.
Attributes sent
2015-08-24T18:23:23.307Z - info: Component registered name=power, type=powerswit
ch.v1.0, cid=f862c0f3-8538-4cad-8792-39e501cf198b, d_id=f2-85-00-49-c6-f5
root@edison:~# systemctl start iotkit-agent
root@edison:~#
```

Действие 3. Запусти Arduino IDE и открой программу A1_B_S1_IoT_Blink.ino.

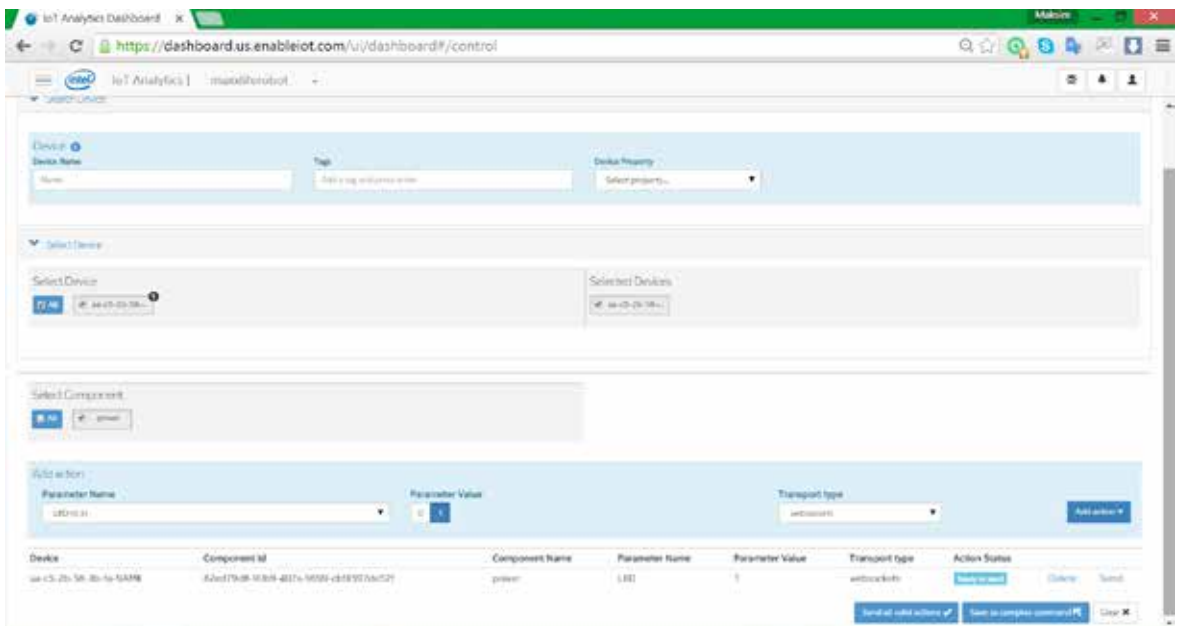
Действие 4. Перед загрузкой проверь код на ошибки.

Действие 5. Загрузи программу на плату.

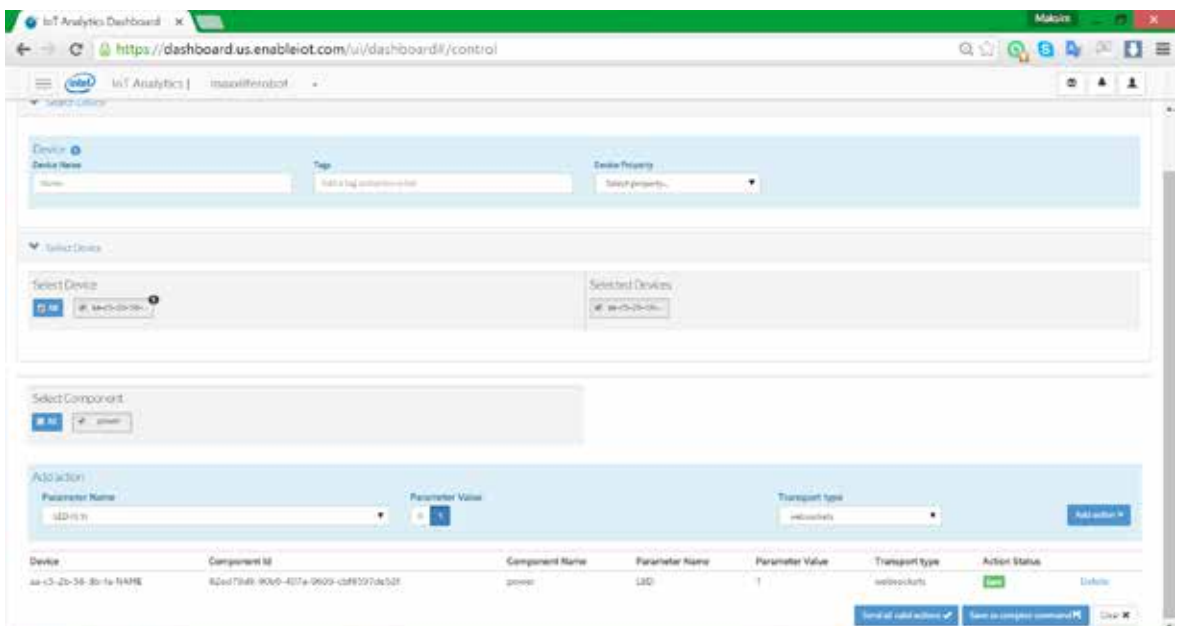
Действие 6. Теперь вернись обратно на портал управления IoT Dashboard и открой вкладку Control.

В Select Device выбери пока единственное свое устройство, в Select Component выбери компонент power, ниже, в Add action, убедись, что стоит LED(0,1).

После этого в Parameter Value выбирай «1» и установи Transport type с параметром websockets, а теперь нажми на кнопку Add action.



Действие 7. Внизу появится твоя сконфигурированная команда, готовая к отправке. Нажми на Send, чтобы отправить команду на плату.



Результат Action Status изменится на `Sent`, а секунд через десять должен загореться светодиод на плате Intel® Edison. Чтобы схема выглядела более эффектно, можно подключить внешний светодиод к 13-му пину на плате! Как это сделать, ты уже знаешь.

Не работает? Открой в Arduino IDE монитор последовательного порта на скорости 115200 бод. Когда будет получена команда, там отобразится информация о ее успешном выполнении.

Перезагрузи плату.

Отправь еще раз команду.

СХЕМА В2 ОТПРАВКА ДАННЫХ С ДАТЧИКА ТЕМПЕРАТУРЫ С ПОМОЩЬЮ ОБЛАКА INTEL® IOT DASHBOARD

Интернет вещей – это не только удаленное управление устройством, но и сбор данных с датчиков и их отправка на облачные сервисы для анализа информации, после чего устройство получает обратную команду с актуальными указаниями. Например, в квартире такой прибор отправляет данные о температуре, влажности, состоянии сигнализации и т. д. на облачный сервис, после чего информация поступает хозяину на смартфон. Он, в свою очередь, может анализировать ее сам либо эту возможность автоматике. Если сработает сигнализация, будет автоматически вызвана служба безопасности либо уведомлен о случившемся хозяин. Возможности интернета вещей безграничны! Идеи за тобой.

Работая над этой схемой, мы прикоснемся к одной из самых динамично развивающихся сфер информационных технологий – будем отправлять данные с датчика температуры (термистора) на облачный сервис IoT Dashboard, который будет автоматически строить график изменения температуры.

Подразумевается, что у тебя открыта Putty и ты вошел в систему Intel® Edison с правами пользователя root, а также, что плата подключена к Wi-Fi.

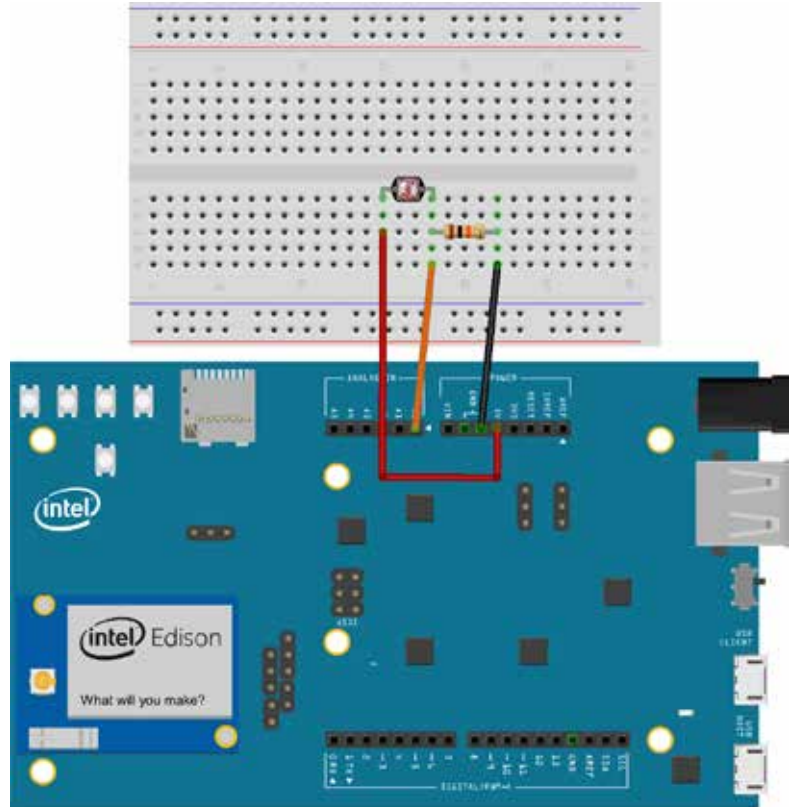
Что необходимо Два микро-USB-кабеля.

Фоторезистор.

Резистор на 10 кОм.

Три провода «папа-папа».

Схема В2



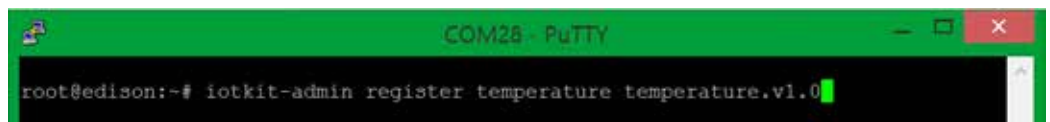
Настройка и сборка схемы

Действие 1. Данная схема очень похожа на Схему А4, за исключением того, что не надо подключать внешний светодиод. Выполни Действия 7–11 из Схемы А4, и у тебя получится нужная тебе Схема В2.

Действие 2. Зайди в консоль и выполни регистрацию датчика температуры.

Команда для регистрации: `iotkit-admin register имя_датчика тип_датчика`

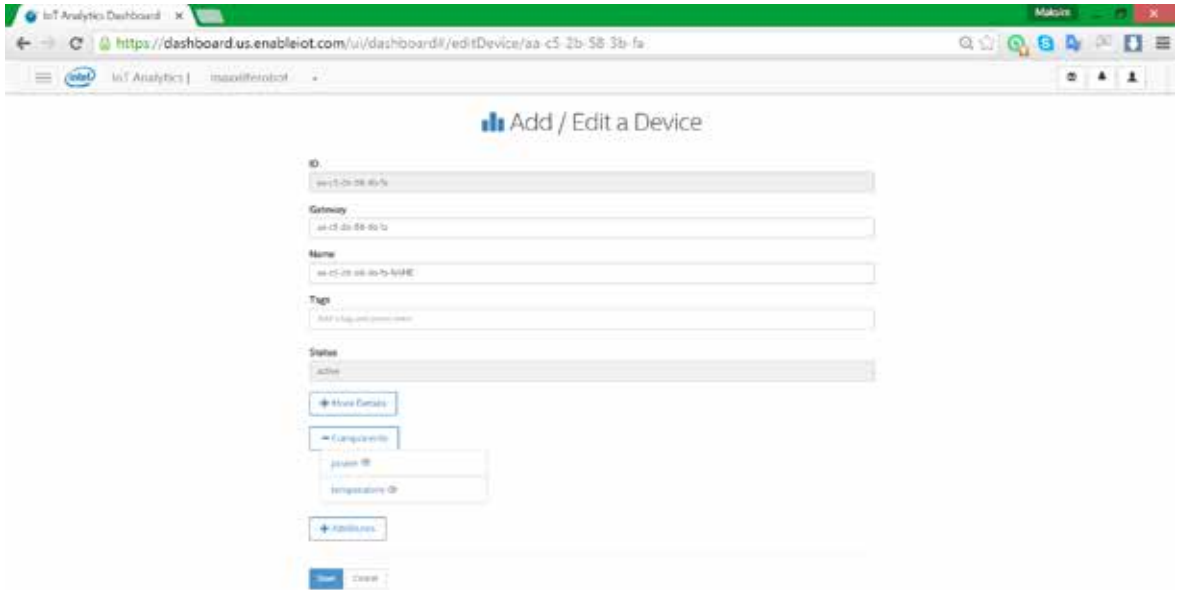
Введи команду для своего случая: `iotkit-admin register temperature temperature.v1.0`



Через некоторое время система уведомит тебя об успешной регистрации датчика.



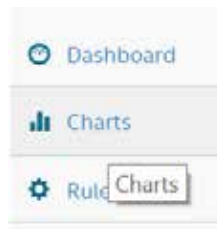
Действие 3. Проверь, появился ли новый датчик на портале. В разделе Add/Edit a Device, пункт Components.



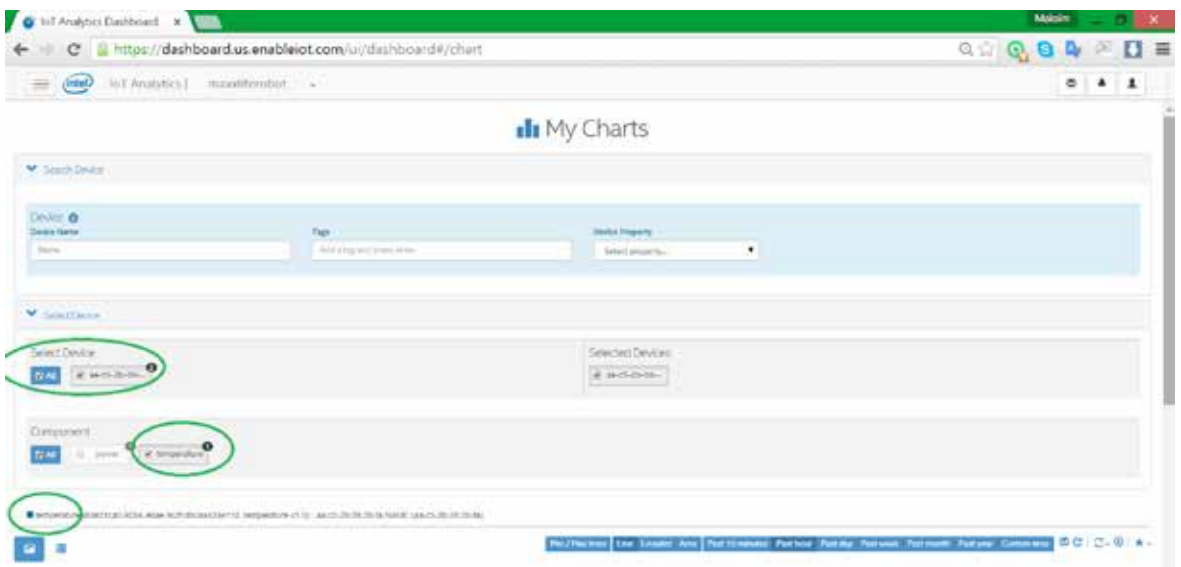
Действие 4. Запусти Arduino IDE и открой программу A1_B_S2_IoT_Temperature.ino.

Действие 5. Загрузи программу на плату.

Действие 6. Теперь вернись обратно на портал управления IoT Dashboard и открой вкладку Charts.



Действие 7. В Select Device выбери пока единственное свое устройство, в Select Component выбери компонент temperature, а ниже убедись, что возле temperature стоит синяя метка.



Результат

Ниже должен появиться график изменения температуры твоего датчика. Ты можешь заходить на портал и изучать информацию с любого устройства, которое подключено к интернету. А главное – ты можешь контролировать показания из любой точки мира!

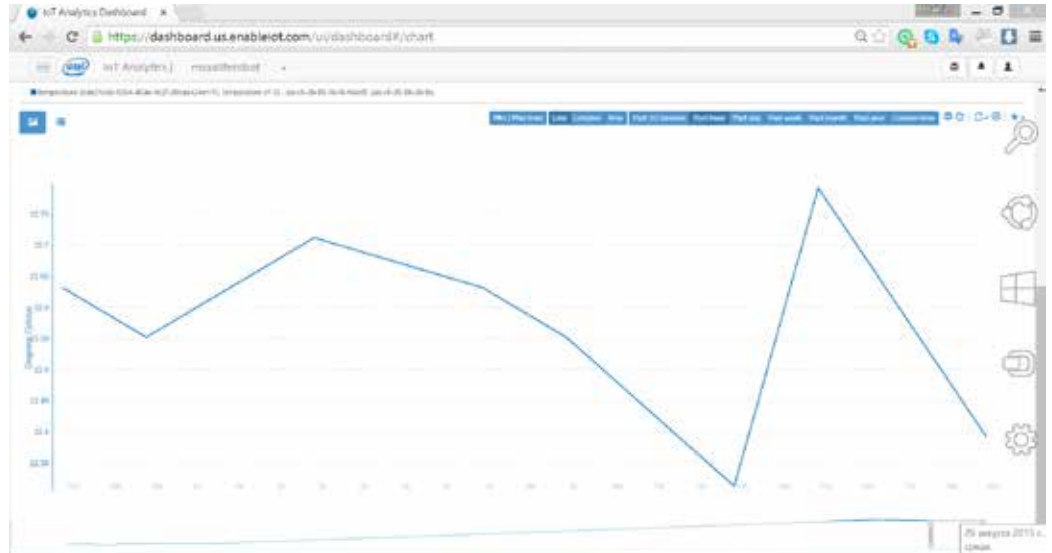


График и его отображение в виде таблицы.

Device	Component Name	Component ID	Component Catalog Type	Timestamp	Value
iot-5-2b-18-0b-5a	temperature	5db271a0-32b4-480e-9c2f-d8ca824e1110	temperature-v1.0	Wed, 26 Aug 2015 19:40:42 GMT+3	24.011178
iot-5-2b-18-0b-5a	temperature	5db271a0-32b4-480e-9c2f-d8ca824e1110	temperature-v1.0	Wed, 26 Aug 2015 19:40:42 GMT+3	22.871428
iot-5-2b-18-0b-5a	temperature	5db271a0-32b4-480e-9c2f-d8ca824e1110	temperature-v1.0	Wed, 26 Aug 2015 19:40:44 GMT+3	24.011178
iot-5-2b-18-0b-5a	temperature	5db271a0-32b4-480e-9c2f-d8ca824e1110	temperature-v1.0	Wed, 26 Aug 2015 19:40:46 GMT+3	22.781588

Предлагаю тебе самостоятельно разобраться в настройках облачного сервиса Intel® IoT Dashboard, чтобы у тебя возникло побольше вопросов, которые ты всегда можешь задать на нашем форуме.

Не работает?

Открой в Arduino IDE монитор последовательного порта на скорости 9600 бод. Там будет отображаться отправляемая в облако температура.

Перезагрузи плату.

Отправь еще раз команду.

ГЛАВА 12

ПОДКЛЮЧЕНИЕ ВНЕШНИХ USB-УСТРОЙСТВ К INTEL® EDISON

В этой главе ты познакомишься с подключением внешних USB-устройств к Intel® Edison. Дополнительная информация и возможные правки к главе доступны по ссылке <http://tiny.cc/a1-edit-c12>.

Сам микрокомпьютер Intel® Edison умеет работать в двух режимах: режиме «хост» (host mode) и режиме «устройство» (device mode). Режимы меняются микропереключателем, который находится между USB-портом типа A (самый большой USB-порт на плате) и двумя микро-USB-портами типа micro-B.

Когда микропереключатель сдвинут в сторону USB-порта типа A, плата работает в режиме «хост», USB-порт A активируется и к нему можно подключать внешние USB-устройства. В данном режиме отключается OTG-порт (центральный микро-USB на плате), поэтому питать плату необходимо от блока питания, который подключается к разъему для питания платы. Также в данном режиме через OTG-порт невозможно загружать на плату Arduino-скетчи т. к. OTG-порт отключен.

Если микропереключатель сдвинут в другую сторону, плата переходит в режим «устройство», USB-порт A выключается, и к нему теперь невозможно подключить внешние USB-устройства. Дополнительно плату можно питать через разъем для питания при помощи блока питания, если ты хочешь обеспечить автономную работу платы. В таком случае можно отключить все USB-кабели от платы и оставить только питание от внешнего блока питания.

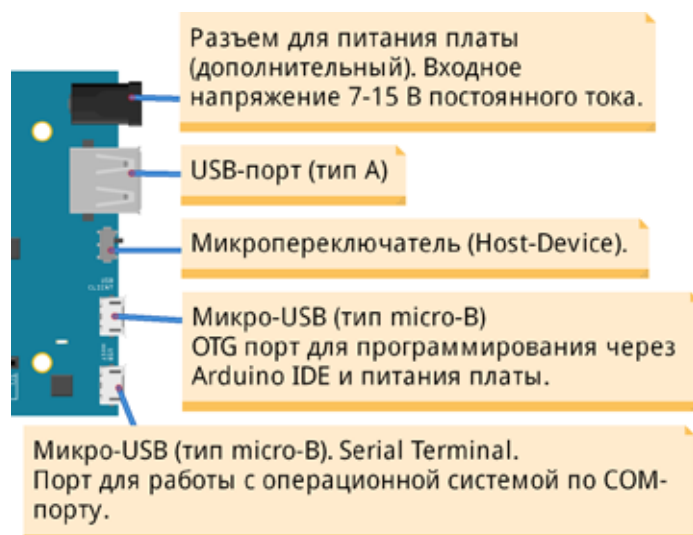


СХЕМА С1 ПОДКЛЮЧЕНИЕ ВЕБ-КАМЕРЫ К INTEL® EDISON. ПОТОКОВОЕ ВЕЩАНИЕ ВИДЕО (ВИДЕОСТРИМИНГ)

Ты сделаешь собственную IP-веб-камеру, которая будет локально транслировать потоковое видео. Таким образом, можно следить за всем, что творится в доме (теперь твоя собака, кот или хомяк всегда будут под контролем), или же сделать робота – и управлять им удаленно.

Собирая схему, ты изучишь текстовый редактор vi, подключение веб-камеры к Intel® Edison, создание программы видеостримера на языке программирования C.

Что необходимо Блок питания на 12 В. Приобрести его ты можешь по ссылке <http://tiny.cc/a1-power-12v>.

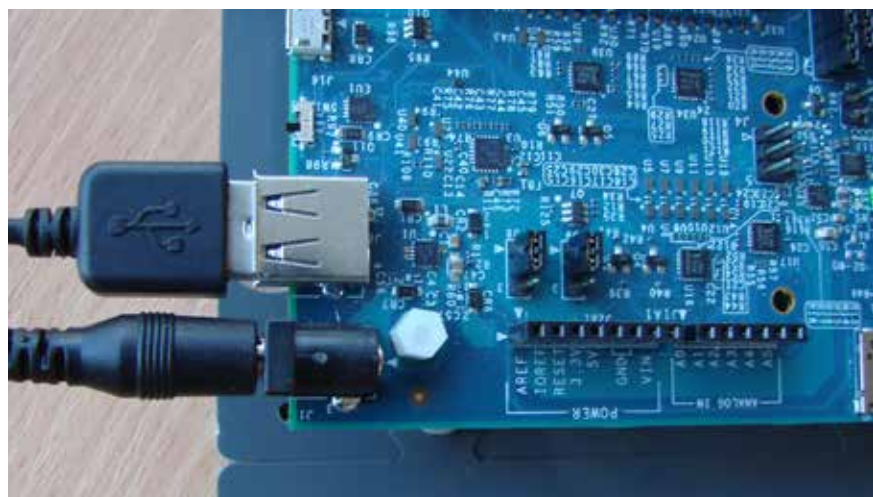
Веб-камера. Ее ты можешь также приобрести по ссылке <http://tiny.cc/a1-web-camera>.

Возможно, у тебя уже есть веб-камера. Она должна быть UVC-совместимой, так что перейди по ссылке <http://www.ideasonboard.org/uvc/> и убедись, подходит ли твоя камера для сборки данной схемы. Желательно, чтобы в строке с моделью камеры стояла зеленая галочка, а не знак предупреждения.

Браузер Mozilla Firefox. Скачать его можно по ссылке <http://tiny.cc/a1-mozilla-download>.

Настройка и сборка схемы

- Действие 1.** Сдвинь микропереключатель в сторону USB-порта типа А.
- Действие 2.** Подключи веб-камеру к USB-порту типа А.
- Действие 3.** Подключи блок питания к розетке, а штекерный разъем блока питания – к разъему питания на плате Intel® Edison.



- Действие 4.** Подсоедини микро-USB-кабель к крайнему микро-USB-порту на плате. После этого подключи Intel® Edison к Wi-Fi (Глава 9, 9.3). Зайди на Intel® Edison через клиент Putty по последовательному порту.
- Действие 5.** После подключения Intel® Edison к Wi-Fi, запомни IP-адрес платы. Узнать его можно с помощью команды `configure_edison -showWiFiIP`
- Действие 6.** Теперь нам необходимо отредактировать текстовый файл, чтобы загрузить пакеты (программы) из репозитория, в котором они хранятся: <http://alextgalileo.altervista.org/edison-package-repo-configuration-instructions.html>.
- Действие 7.** Чтобы отредактировать нужный файл, используй текстовый редактор `vi`.

Текстовый редактор `vi` отличается от привычного «Блокнота». В нем управление осуществляется только английскими символами с клавиатуры, а также клавишами управления курсором (вправо-влево).

Команды для управления текстовым редактором `vi`

Полный список команд и подробное описание редактора ты можешь найти здесь: <http://tiny.cc/a1-vi-editor>.

`vi filename` – редактировать файл с именем `filename`.

`vi newname` – создать файл с именем `newname`, если такого файла не существует.

Чтобы использовать управляющий символ, после редактирования текста необходимо нажать `Esc`, а потом нужный символ.

`x` – удалить символ.

`i` – вставить текст слева от курсора.

`a` – вставить текст справа от курсора.

`h` – переместить курсор налево на один пробел.

`j` – переместить курсор вниз на одну строку.

`k` – переместить курсор вверх на одну строку.

`l` – переместить курсор вправо на один пробел.

`o` – вставить пустую строку под курсором.

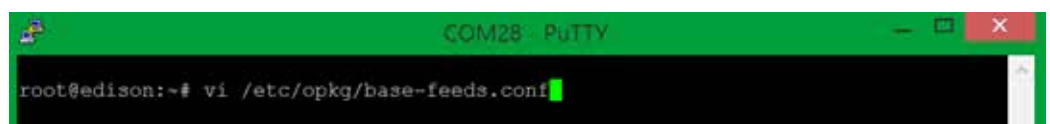
`ZZ` – сохранить файл и выйти из редактора.

`:w` – сохранить файл.

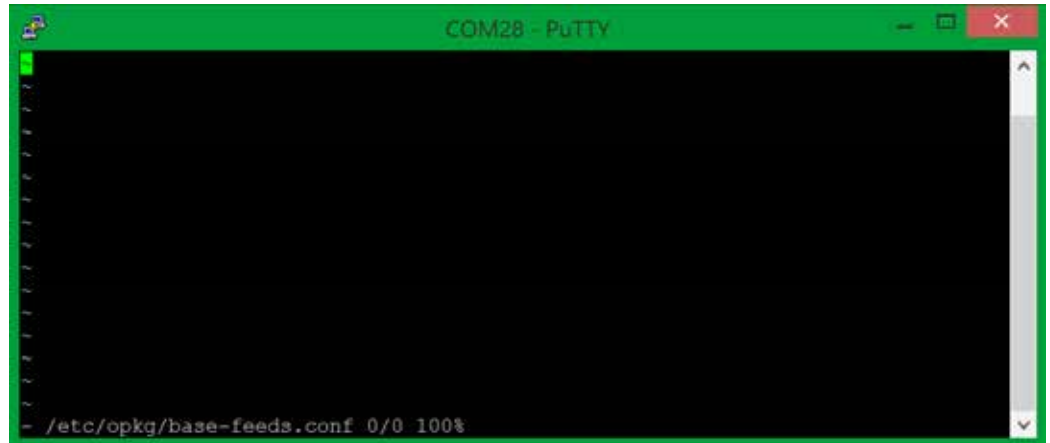
`:q!` – отменить изменения и выйти из редактора.

- Действие 8.** Открой файл при помощи редактора `vi`. Введи команду `vi /etc/opkg/base-feeds.conf`

После чего нажми `Enter`.



Действие 9. Откроется или автоматически создастся файл `base-feeds.conf`.



Действие 10. Теперь надо модифицировать данный файл. Нажми английский символ «x». Потом символ «i».

При наборе строк следи за межбуквенным расстоянием и пробелами, чтобы не пропустить пробел в нужном месте.

Действие 11. Набираем первую строку: `src/gz all http://repo.opkg.net/edison/repo/all`



В конце строки ждем Esc, а после символ «o», который переведет курсор на одну строку вниз.



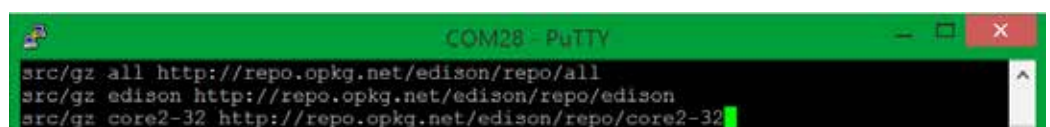
Действие 12. Набираем вторую строку: `src/gz edison http://repo.opkg.net/edison/repo/edison`



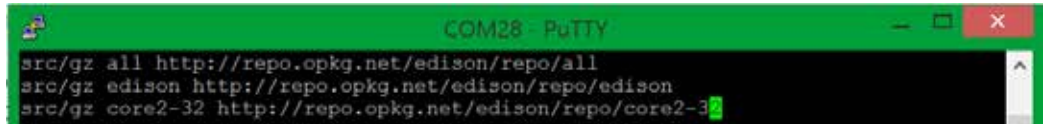
В конце строки ждем Esc, а после символ «o», который переведет курсор опять на одну строку вниз.



Действие 13. Набери третью строку: `src/gz core2-32 http://repo.opkg.net/edison/repo/core2-32`



В конце строки жмем `Esc`, а потом два раза заглавный символ «`Z`».



```
COM28 - PuTTY
src/gz all http://repo.opkg.net/edison/repo/all
src/gz edison http://repo.opkg.net/edison/repo/edison
src/gz core2-32 http://repo.opkg.net/edison/repo/core2-32
```

Действие 14. В Шаге 13 ты вышел из текстового файла. Набери `opkg update` и нажми `Enter`.



```
COM28 - PuTTY
root@edison:~# vi /etc/opkg/base-feeds.conf
root@edison:~# opkg update
```

Если ты ввел текст верно, начнется скачивание необходимых файлов.



```
COM28 - PuTTY
root@edison:~# vi /etc/opkg/base-feeds.conf
root@edison:~# opkg update
Downloading http://repo.opkg.net/edison/repo/all/Packages.gz.
```

Действие 15. Теперь необходимо установить `git`. `Git` – это распределённая система управления версиями. Введи команду для установки пакета `opkg install git`



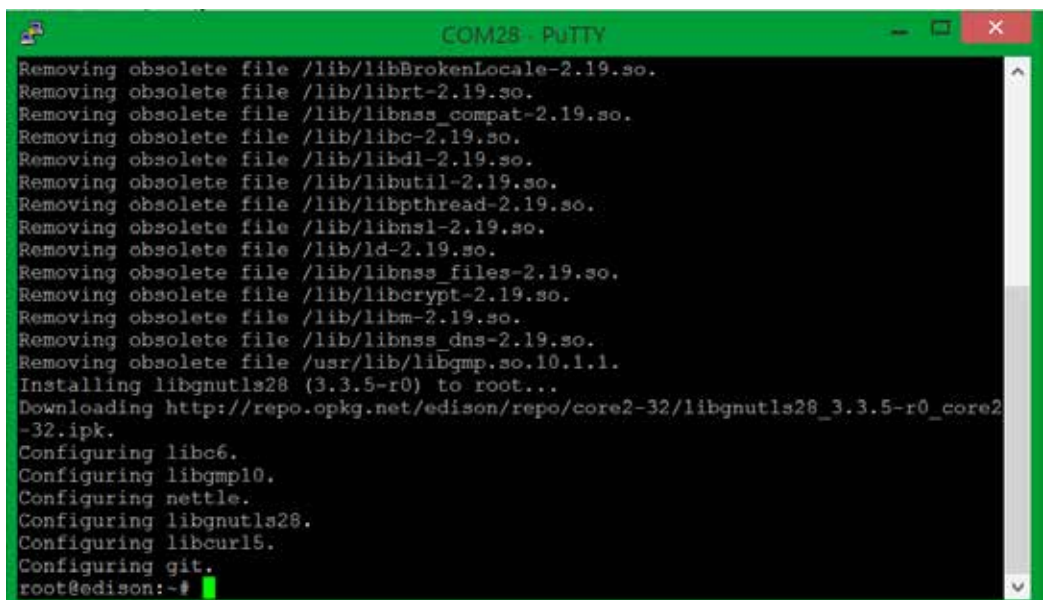
```
COM28 - PuTTY
root@edison:~# opkg install git
```

Начнется скачивание пакета `git` и его установка в системе.



```
COM28 - PuTTY
Installing git (2.0.1-r0) to root...
Downloading http://repo.opkg.net/edison/repo/core2-32/git_2.0.1-r0_core2-32.ipk.
```

Скачивание занимает 2–3 минуты. Когда программа установится, на мониторе должно быть написано что-то вроде этого.



```
COM28 - PuTTY
Removing obsolete file /lib/libBrokenLocale-2.19.so.
Removing obsolete file /lib/libbrt-2.19.so.
Removing obsolete file /lib/libnss_compat-2.19.so.
Removing obsolete file /lib/libc-2.19.so.
Removing obsolete file /lib/libdl-2.19.so.
Removing obsolete file /lib/libutil-2.19.so.
Removing obsolete file /lib/libpthread-2.19.so.
Removing obsolete file /lib/libnsl-2.19.so.
Removing obsolete file /lib/ld-2.19.so.
Removing obsolete file /lib/libnss_files-2.19.so.
Removing obsolete file /lib/libcrypt-2.19.so.
Removing obsolete file /lib/libm-2.19.so.
Removing obsolete file /lib/libnss_dns-2.19.so.
Removing obsolete file /usr/lib/libgmp.so.10.1.1.
Installing libgnutls28 (3.3.5-r0) to root...
Downloading http://repo.opkg.net/edison/repo/core2-32/libgnutls28_3.3.5-r0_core2-32.ipk.
Configuring libc6.
Configuring libgmp10.
Configuring nettle.
Configuring libgnutls28.
Configuring libcurl5.
Configuring git.
root@edison:~#
```

Действие 16. Теперь скопируй исходники видеостримера на Intel® Edison. Введи команду `git clone https://github.com/mrobotby/A1_video_stream` и нажми Enter. На Intel® Edison будет загружен репозиторий с исходниками программы для видеостримера.

```
COM28 - PuTTY
root@edison:~# git clone https://github.com/mrobotby/A1_video_stream
```

В конце копирования окно должно выглядеть приблизительно так.

```
COM28 - PuTTY
root@edison:~# git clone https://github.com/mrobotby/A1_video_stream
Cloning into 'A1_video_stream'...
remote: Counting objects: 14, done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 14 (delta 1), reused 14 (delta 1), pack-reused 0
Unpacking objects: 100% (14/14), done.
Checking connectivity... done.
root@edison:~#
```

Действие 17. Теперь введи `lsmod | grep uvc` и нажми Enter. Если все так, как на экране, или похоже, то переходи к следующему шагу.

```
COM28 - PuTTY
root@edison:~# lsmod | grep uvc
uvcvideo                71516  0
videobuf2_vmalloc      13003  1 uvcvideo
videobuf2_core         37707  1 uvcvideo
root@edison:~#
```

Действие 18. Проверь веб-камеру на видимость в системе командой `ls -l /dev/video0` и нажми Enter.

```
COM28 - PuTTY
root@edison:~# ls -l /dev/video0
crw-rw----  1 root  video   81,  0 Aug 29 14:33 /dev/video0
root@edison:~#
```

Действие 19. Теперь установи дополнительную библиотеку командой `opkg install libv4l-dev`

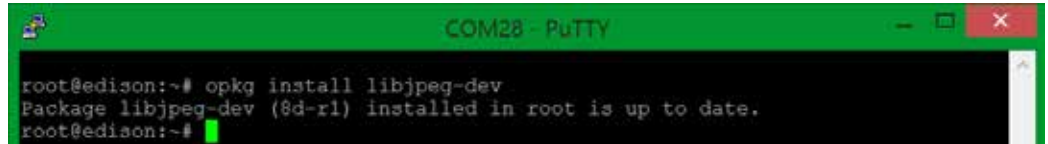
```
COM28 - PuTTY
root@edison:~# opkg install libv4l-dev
```

```
COM28 - PuTTY
root@edison:~# opkg install libv4l-dev
Installing libv4l-dev (1.0.1-r0) to root...
Downloading http://repo.opkg.net/edison/repo/core2-32/libv4l-dev_1.0.1-r0_core2-32.ipk.
Installing libv4l (1.0.1-r0) to root...
Downloading http://repo.opkg.net/edison/repo/core2-32/libv4l_1.0.1-r0_core2-32.ipk.
Installing libjpeg-dev (8d-r1) to root...
Downloading http://repo.opkg.net/edison/repo/core2-32/libjpeg-dev_8d-r1_core2-32.ipk.
Configuring libv4l.
Configuring libjpeg-dev.
Configuring libv4l-dev.
root@edison:~#
```


Действие 20. Еще одна библиотека устанавливается командой `opkg install libjpeg-dev`

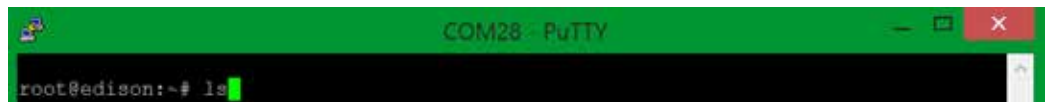


```
root@edison:~# opkg install libjpeg-dev
```



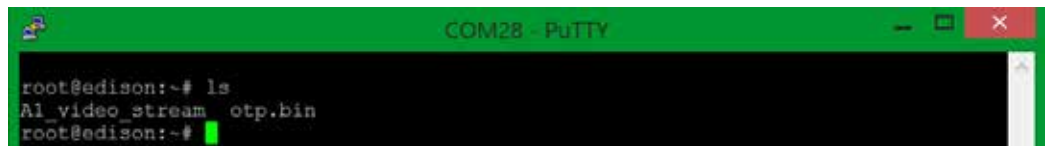
```
root@edison:~# opkg install libjpeg-dev
Package libjpeg-dev (8d-r1) installed in root is up to date.
root@edison:~#
```

Действие 21. Теперь введи команду `ls`, чтобы увидеть, какие файлы и папки находятся в нашей директории Intel® Edison.



```
root@edison:~# ls
```

Видишь `A1_video_stream`? Исходники программы успешно скопированы на компьютер.



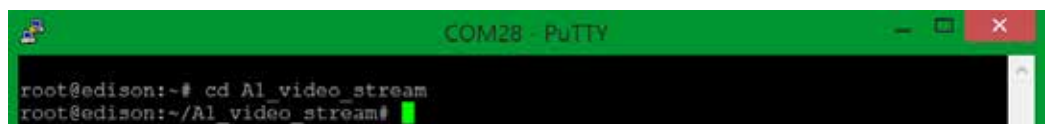
```
root@edison:~# ls
A1_video_stream otp.bin
root@edison:~#
```

Действие 22. Теперь надо зайти в директорию `A1_video_stream`. Введи команду `cd A1_video_stream` и нажми Enter.



```
root@edison:~# cd A1_video_stream
```

Ты перешел в директорию `A1_video_stream`.



```
root@edison:~# cd A1_video_stream
root@edison:~/.A1_video_stream#
```

Действие 23. Командой `ls` посмотри файлы в данной директории. Это исходники, или source-файлы, будущей программы. Для того чтобы она работала, ее надо «собрать».



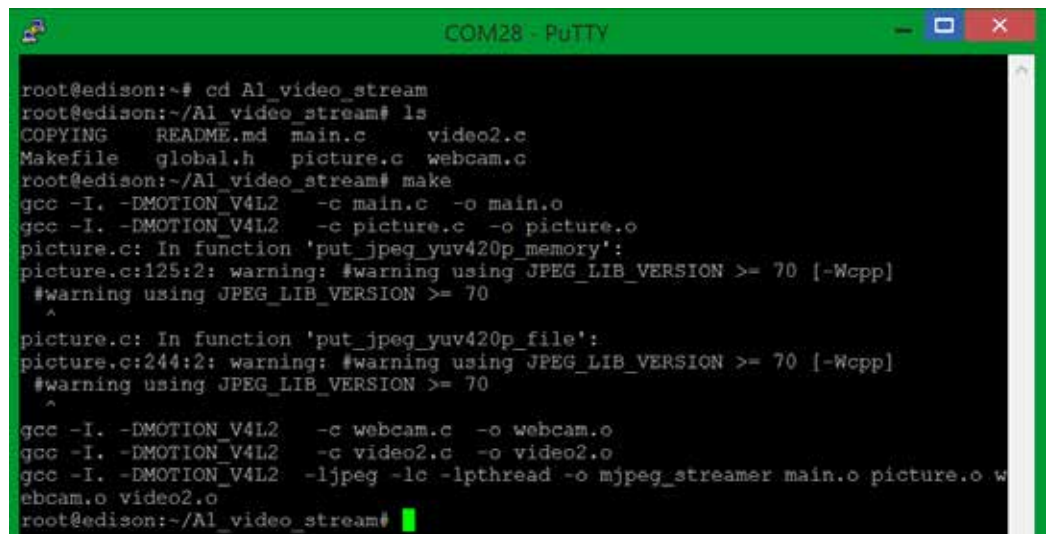
```
root@edison:~# cd A1_video_stream
root@edison:~/.A1_video_stream# ls
COPYING  README.md  main.c    video2.c
Makefile global.h  picture.c webcam.c
root@edison:~/.A1_video_stream#
```

Действие 24. Введи команду `make` и нажми Enter. После этого начнется сборка программы.



```
root@edison:~# cd A1_video_stream
root@edison:~/.A1_video_stream# ls
COPYING  README.md  main.c    video2.c
Makefile global.h  picture.c webcam.c
root@edison:~/.A1_video_stream# make
```

После сборки программы окно должно выглядеть так.



```

COM28 - PuTTY
root@edison:~# cd A1_video_stream
root@edison:~/A1_video_stream# ls
COPYING  README.md  main.c    video2.c
Makefile global.h  picture.c webcam.c
root@edison:~/A1_video_stream# make
gcc -I. -DMOTION_V4L2 -c main.c -o main.o
gcc -I. -DMOTION_V4L2 -c picture.c -o picture.o
picture.c: In function 'put_jpeg_yuv420p_memory':
picture.c:125:2: warning: #warning using JPEG_LIB_VERSION >= 70 [-Wcpp]
#warning using JPEG_LIB_VERSION >= 70
^
picture.c: In function 'put_jpeg_yuv420p_file':
picture.c:244:2: warning: #warning using JPEG_LIB_VERSION >= 70 [-Wcpp]
#warning using JPEG_LIB_VERSION >= 70
^
gcc -I. -DMOTION_V4L2 -c webcam.c -o webcam.o
gcc -I. -DMOTION_V4L2 -c video2.c -o video2.o
gcc -I. -DMOTION_V4L2 -ljpeg -lc -lpthread -o mjpeg_streamer main.o picture.o webcam.o video2.o
root@edison:~/A1_video_stream#

```

Действие 25. Теперь, чтобы запустить программу, необходимо набрать в директории `A1_video_stream` команду:

```
./mjpeg_streamer
```

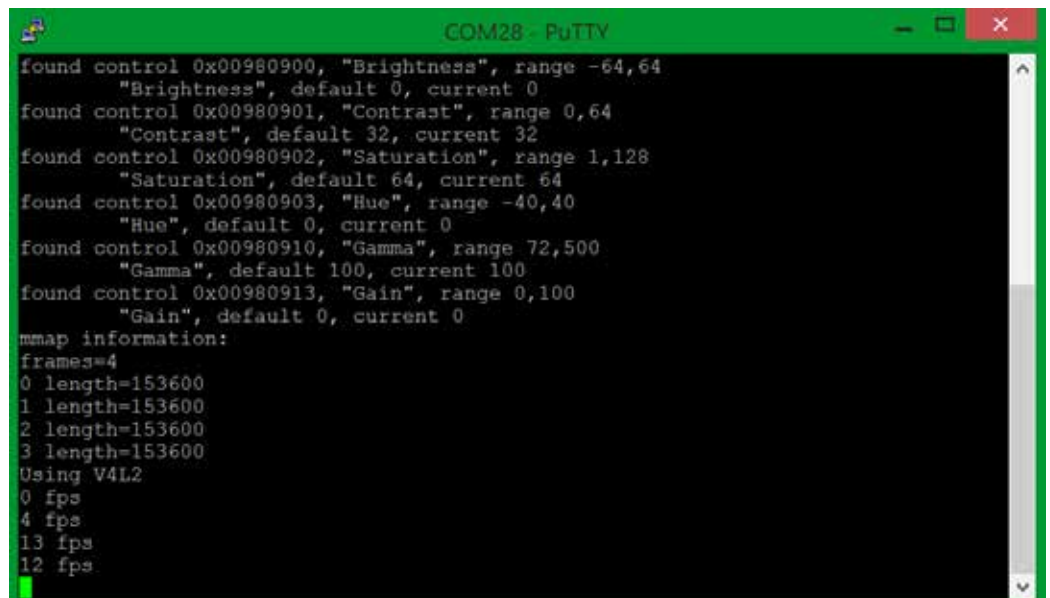


```

COM28 - PuTTY
root@edison:~/A1_video_stream# ./mjpeg_streamer

```

Программа начнет выполняться, и начнет меняться параметр `fps`.



```

COM28 - PuTTY
found control 0x00980900, "Brightness", range -64,64
  "Brightness", default 0, current 0
found control 0x00980901, "Contrast", range 0,64
  "Contrast", default 32, current 32
found control 0x00980902, "Saturation", range 1,128
  "Saturation", default 64, current 64
found control 0x00980903, "Hue", range -40,40
  "Hue", default 0, current 0
found control 0x00980910, "Gamma", range 72,500
  "Gamma", default 100, current 100
found control 0x00980913, "Gain", range 0,100
  "Gain", default 0, current 0
mmap information:
frames=4
0 length=153600
1 length=153600
2 length=153600
3 length=153600
Using V4L2
0 fps
4 fps
13 fps
12 fps

```

Чтобы приостановить выполнение программы, необходимо нажать сочетание клавиш `Ctrl+Z`. Для продолжения выполнения программы нужно набрать команду `fg` и нажать `Enter`. Чтобы закрыть программу, следует нажать сочетание клавиш `Ctrl+C`. Если надо запустить программу, введи опять команду:

```
./mjpeg_streamer
```

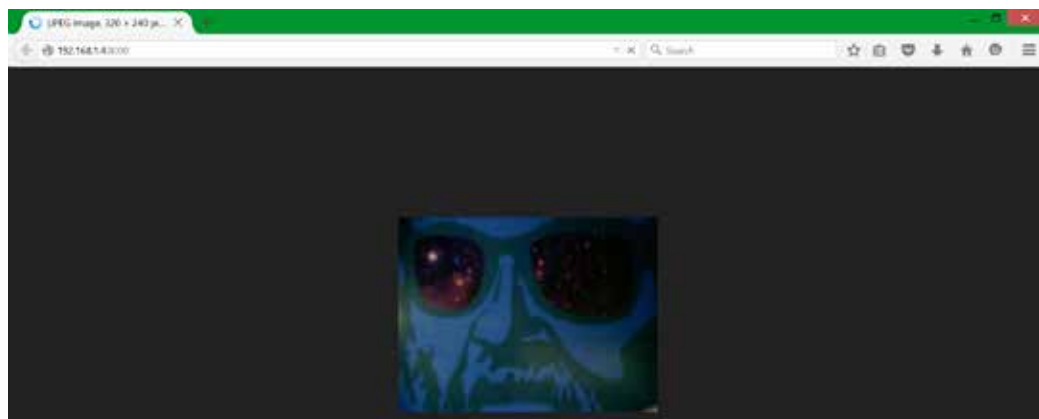
Действие 26. Теперь открой браузер Mozilla Firefox и в адресной строке набери IP-адрес платы, после него через двоеточие номер порта 8090.

Пример: 192.168.1.4:8090

И нажми `Enter`.

Результат

В окне страницы браузера должно появиться изображение с камеры. Убедись, что твой компьютер подключен к той же сети, что и Intel® Edison. Ты можешь установить Mozilla Firefox на свой смартфон и смотреть с него потоковое видео!



Не работает?

Если на каком-либо шаге возникли проблемы, то незамедлительно обращайся в нашу службу поддержки с подробным описанием своих действий.

ГЛАВА 13

ОШИБКИ INTEL® EDISON

И СПОСОБЫ ИХ УСТРАНЕНИЯ

Если твоей ошибки здесь нет либо ты не смог с ней справиться, пиши нам службу поддержки или поделись своей проблемой на форуме.

13.1. Скетч не загружается на плату

Известная проблема, из-за которой можно потерять очень много времени. Корень данной проблемы – твоя невнимательность.

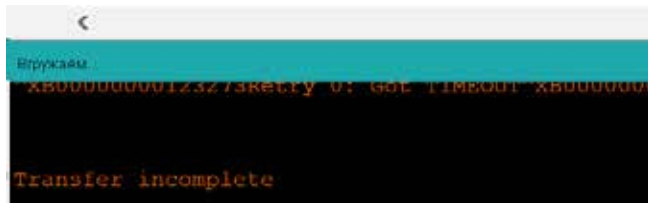
Каждый раз, когда ты открываешь Arduino IDE, тебе необходимо проверить на вкладке «Инструменты», выбрана ли плата Intel® Edison в разделе «Плата», а также выбран ли нужный COM-порт в разделе «Порт».

Ты допустил ошибку в коде. Проверь на парность фигурные скобки, наличие точек с запятой; в программе используются библиотеки, которые не поддерживаются платой Intel® Edison.

13.2. Arduino IDE не видит нужный COM-порт

Такое действительно может быть, если ты неправильно установил драйвера либо же Intel® Edison еще не загрузился. Через пару минут после подключения Intel® Edison через OTG-порт к компьютеру, нужный порт должен появиться. Если нет, то отсоедини плату и подсоедини обратно. Подожди пару минут. Если по-прежнему ничего не происходит, отсоедини плату и перезагрузи компьютер. После перезагрузки подсоедини плату, запусти среду Arduino IDE. Нужный порт должен появиться.

13.3. При загрузке скетча Arduino IDE пишет Transfer incomplete

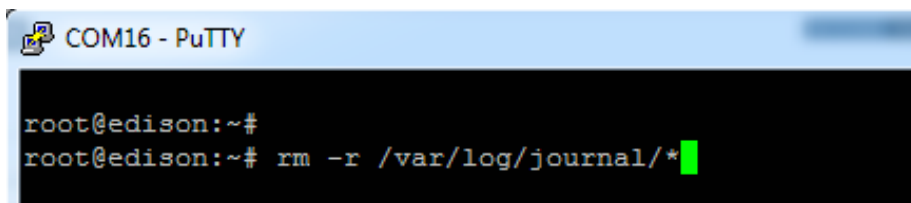


Проблема легко решается, если ты разобрался с [Главой 10](#). Если ты еще не дошел до [Главы 10](#), а проблема есть, то делай все, как здесь написано. Прочитай [Главы 10, 10.1.2](#), выполни установку Putty. Потом в [Главе 10](#) найди Инструкцию № 4, выполни только Шаг 2, Шаги 11–15. В окне консоли тебе необходимо именно набрать данную строку `rm -r /var/log/journal/*`

Пробелы есть между `rm -r`

Пробелы есть между `-r /var/log/journal/*`

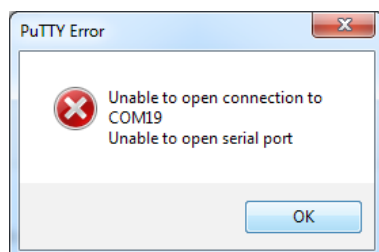
Больше нигде пробелов нет.



После чего нажми Enter. И попробуй опять загрузить программу.

13.4. Скетч долго компилируется в Arduino IDE

Такое возможно, если у тебя есть антивирус (Avast может создавать проблемы) или другие программы, которые мешают Arduino IDE. Обычно компиляция происходит мгновенно, но может занимать до трех минут. Это нормально. Однако, если ты не любишь ждать, попробуй отключить антивирус. Если это помогло, значит, дело в нем.



13.5. Putty не открывает нужный COM-порт

Во-первых, тебе надо убедиться, что ты открываешь нужный COM-порт. Во-вторых, необходимо немного подождать (2–3 минуты), чтобы Intel® Edison загрузился и нужные порты появились в системе.

ГЛАВА 14

КЛАССНЫЕ ПРОЕКТЫ ДЛЯ ТВОЕГО ВДОХНОВЕНИЯ

Ты осилил целую книгу и даже ничего не спалил, я надеюсь. А теперь я предлагаю твоему вниманию подборку самых интересных, на мой взгляд, проектов. Специально для тебя у Intel® есть сайт iq.intel.com, на котором представлены все передовые решения и идеи от Intel®. Вводи в поиске слово Edison, и сразу увидишь лучшие проекты на плате Intel® Edison. Не исключено, что и твой проект окажется там!

В поисках источника вдохновения заходи на сайт <http://www.instructables.com/id/Intel-Edison-Projects/>. Там ты найдешь много разных проектов с подробными инструкциями, как повторить каждый из них. Не забывай читать англоязычный форум на официальном сайте Intel® <http://communities.intel.com/community/tech/edison>. Создавай темы с вопросами, и тебе обязательно ответят профессиональные разработчики. Только для начала тебе необходимо будет зарегистрироваться по ссылке <https://communities.intel.com/create-account.jspa>. Знакомься, общайся, тебе всегда помогут. Если ты не знаешь, как что-то сделать, всегда есть человек, который знает, и ты его всегда найдешь.

Предлагаю тебе посетить нашу [maxxlife] «Кухню роботов» и познакомиться интересными людьми и проектами! Подробнее о нашей лаборатории по робототехнике и электронике ты можешь узнать от участников группы «ВКонтакте» www.vk.com/cookrobot и на сайтах www.mrobot.by и www.lab.mrobot.by. И не забывай рассказывать о своих проектах на нашем форуме www.forum.mrobot.by.

ЭПИЛОГ

Вот и закончилось твое первое путешествие в мир электроники, программирования и робототехники. На мой взгляд, главное – это не конечный результат, а путь, который ты преодолел. За это время ты многое узнал, многому научился и многое понял: понял, как разбираться в электронных компонентах, знаешь, чем термистор отличается от фоторезистора, имеешь представление о том, как работают многие устройства в нашем мире, знаешь и понимаешь, как собирать электрические схемы и цепи, умеешь программировать микрокомпьютер Intel® Edison и работать с ОС Linux, понял, что такое интернет вещей и многое другое. Но главное – ты приобрел бесценные навыки! Теперь осталось эти навыки шлифовать и расширять в кругу таких же изобретателей и энтузиастов, каким стал ты! Ведь знания и навыки, которые ты получил, подобны кипящей воде. Если не поддерживать огонь, она становится прохладной.

До скорого!

О КОРПОРАЦИИ INTEL

Корпорация Intel – ведущий мировой производитель инновационных полупроводниковых компонентов — разрабатывает технологии, продукцию и инициативы, направленные на постоянное повышение качества жизни людей и совершенствование методов их работы. Intel, логотип Intel являются товарными знаками корпорации Intel в США и других странах.

Дополнительную информацию о корпорации Intel можно найти на веб-сайте www.intel.ru/pressroom и на сайтах:

<http://www.intel.ru/>

<http://iq.intel.ru/>

<https://edugalaxy.intel.ru/>

<http://stemcentre.ru/>

