

Электроника

Виктор Петин



Arduino и Raspberry Pi в проектах Internet of Things



Материалы
на www.bhv.ru

- исходные коды скетчей Arduino из книги
- исходные коды библиотек Arduino
- исходные коды python-скриптов для проектов с Raspberry Pi

Интернет вещей — будущее,
которое уже наступило. Не опоздай!

УДК 004.4
ББК 32.973.26-018.2
П29

Петин В. А.

П29 Arduino и Raspberry Pi в проектах Internet of Things. — СПб.:
БХВ-Петербург, 2016. — 320 с.: ил. — (Электроника)
ISBN 978-5-9775-3646-2

Рассмотрено создание простых устройств в рамках концепции Интернета вещей (IoT, Internet of Things) на базе популярной платформы Arduino и микрокомпьютера Raspberry Pi. Показана установка и настройка среды разработки приложений Arduino IDE, а также среда макетирования Frizing. Описаны технические возможности, особенности подключения и взаимодействия различных датчиков и исполнительных устройств. Показана организация доступа разрабатываемых проектов к сети Интернет, отправка и получение ими данных с использованием популярных облачных IoT сервисов: Narodmon, ThingSpeak, Xively, Weaved, Blynk, Wylidrin и др. Уделено внимание обмену данными с помощью платы GPRS/GSM Shield. Рассмотрен проект создания собственного сервера для сбора по сети данных с различных устройств на платформе Arduino. Показано как использовать фреймворк WebIOPI для работы с Raspberry Pi. Приведены примеры использования Wi-Fi-модуля ESP8266 в проектах «Умный дом». На сайте издательства размещен архив с исходными кодами программ и библиотек.

Для интересующихся современной электроникой

УДК 004.4
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Екатерина Капальгина</i>
Редактор	<i>Григорий Добин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Марины Дамбиевой</i>

Подписано в печать 12.01.16.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 25,8.
Тираж 1200 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

Оглавление

Глава 1. Интернет вещей (вместо введения)	7
Глава 2. Среда программирования Arduino IDE	9
2.1. Установка Arduino IDE	10
2.1.1. В ОС Windows	10
2.1.2. В ОС Linux	12
2.1.3. В Mac OS X	13
2.2. Настройка Arduino IDE	13
Глава 3. Среда разработки Fritzing	17
3.1. Загрузка и установка среды Fritzing	17
3.2. Главное окно среды Fritzing	17
3.3. Создание схемы соединений	20
3.4. Создание принципиальной схемы	21
3.5. Добавление компонентов в среду Fritzing	22
Глава 4. Arduino и аналоговые датчики	25
4.1. Аналоговые датчики (сенсоры)	25
4.2. Arduino и датчик температуры LM335	27
4.3. Arduino, Ethernet Shield/W5100 и облачные сервисы	29
4.3.1. Отправка данных на сайт «Народный мониторинг»	33
4.3.2. Чтение данных с фоторезистора	39
4.3.3. Отправка данных в сервис ThingSpeak	41
4.4. Arduino и инфракрасные датчики расстояния SHARP	49
4.4.1. Подключение датчиков Sharp к Arduino	51
4.4.2. Подсчет количества посетителей магазина	53
4.4.3. Приложение ThingTweet сервиса ThingSpeak	55
4.4.4. Отправка данных о количестве посетителей в Twitter из Arduino	57
Глава 5. Использование Arduino в качестве контроллера исполнительных устройств	63
5.1. Arduino и электромагнитное реле	63
5.1.1. Электромагнитное реле	63
5.1.2. Устройство и принцип работы электромагнитного реле	64
5.1.3. Подключение реле к Arduino	65

5.2. Arduino и твердотельное реле.....	67
5.3. Arduino и диммер.....	68
5.3.1. Диммер.....	68
5.3.2. Подключение диммера к Arduino.....	69
5.3.3. Скетч управления диммером.....	70
5.4. Arduino и сервоприводы.....	72
5.4.1. Принципы управления сервоприводами.....	73
5.4.2. Управление сервоприводом с помощью Arduino.....	75
5.5. Arduino и библиотека TinyWebServer.....	77
5.5.1. Использование файлов с SD-карты для формирования веб-страниц.....	78
5.5.2. Включение/выключение реле с веб-страницы.....	79
5.5.3. Веб-страница для управления реле.....	80
5.5.4. Веб-страница для управления сервоприводом.....	85

Глава 6. Arduino и устройства I²C 89

6.1. Обзор протокола I ² C.....	89
6.2. Arduino и библиотека Wire.....	93
6.3. Arduino и датчик освещенности BH1750 на шине I ² C.....	96
6.4. Arduino и сервис Xively.....	99
6.4.1. Отправка данных в сервис Xively.....	102
6.4.2. Получение данных из сервиса Xively.....	105
6.5. Arduino и датчик влажности и температуры SHT21 на шине I ² C.....	107
6.6. Arduino и сервис Xively (продолжение).....	110
6.6.1. Отправка мультиданных в сервис Xively.....	110
6.6.2. Получение мультиданных из сервиса Xively.....	113
6.7. Arduino и часы реального времени на шине I ² C.....	116
6.8. Arduino и SD-карта: чтение и запись данных.....	121

Глава 7. Arduino и 1-Wire 125

7.1. Технология 1-Wire.....	125
7.2. Применение 1-Wire.....	128
7.3. Интерфейс 1-Wire.....	129
7.3.1. Обмен информацией по шине 1-Wire.....	130
7.3.2. Протокол обмена информацией 1-Wire.....	133
7.4. Arduino и цифровой датчик температуры DS18B20.....	135
7.4.1. Цифровой датчик температуры DS18B20.....	135
7.4.2. Использование библиотеки OneWire для получения данных температуры с датчика DS18B20.....	138

Глава 8. Сервер для сбора данных с Ethernet-модулей датчиков, установленных на Arduino 141

8.1. Датчики влажности DHT11 и DHT22.....	141
8.1.1. Подключение датчиков DHT к Arduino.....	143
8.1.2. Библиотека DHT.....	143
8.2. Модуль датчика движения HC-SR501.....	145
8.3. Модуль датчика звука FC-04.....	148
8.4. Ethernet-модуль датчиков на Arduino.....	149
8.5. Сервер сбора данных.....	153

Глава 9. Обмен данными с помощью платы GPRS/GSM Shield.....	157
9.1. Отправка и получение SMS-сообщений	159
9.2. Отправка данных на сайт «Народный мониторинг».....	162
Глава 10. Проект Blynk: управление Arduino с планшета.....	169
10.1. Начало работы: тестовый пример	170
10.2. Управление с планшета исполнительными устройствами, подключенными к Arduino.....	178
10.3. Отправка данных из Arduino на экран планшета	181
Глава 11. Микрокомпьютер Raspberry Pi.....	187
11.1. Технические характеристики и возможности Raspberry Pi.....	188
11.2. Установка операционной системы	191
11.3. Первоначальная настройка ОС Raspbian	194
11.3.1. Меню конфигурации	194
11.3.2. Настройка сетевых параметров.....	196
11.3.3. Настройка доступа по Wi-Fi.....	196
11.3.4. Подключение 3G-модема	199
11.4. Интерфейс GPIO	203
11.4.1. Управление GPIO из оболочки bash	206
11.4.2. Управление GPIO командами языка Python	206
11.5. Raspberry Pi и датчик температуры DS18B20 на шине 1-Wire	209
11.5.1. Подключение датчика DS18B20 к Raspberry Pi.....	209
11.5.2. Отправка данных с датчика DA18B20 в сервис «Народный мониторинг»	212
11.6. Raspberry Pi и датчик освещенности BH1750 на шине I ² C	215
11.6.1. Подключение датчика BH1750 к Raspberry Pi.....	215
11.6.2. Получение на Raspberry Pi данных с датчика BH1750.....	217
Глава 12. WebIOPi — веб-интерфейс и облако для Raspberry Pi.....	219
12.1. Установка WebIOPi на ОС Raspbian	219
12.2. Задание пользовательского пароля WebIOPi	221
12.3. Настройка сервера WebIOPi	222
12.4. Javascript-библиотека webiopi.js	223
12.4.1. Функции библиотеки webiopi.js	224
12.5. Проект управления веб-камерой на сервоприводах	230
12.6. WebIOPi — подключение устройств	236
12.7. Доступ к устройству из сервиса Weaved	238
12.7.1. Установка сервиса Weaved.....	238
12.7.2. Подключение к Raspberry Pi в сервисе Weaved.....	242
Глава 13. Проект Wyliodrin: управление удаленными устройствами из браузера	245
13.1. Добавление устройства в профиль	246
13.2. Запись образа Wyliodrin на SD-карту.....	247
13.2.1. ...в ОС Windows	248
13.2.2. ...в ОС Linux.....	249
13.2.3. ...в Mac OS.....	250
13.2.4. ...в ОС Raspbian	250
13.3. Запись на SD-карту настроек Wyliodrin.....	250

13.4. Подключение Raspberry Pi к Wyliodrin.....	251
13.5. Создание приложения в графической среде программирования	253
13.6. Включение/выключение светодиода с веб-страницы	257
13.7. Подключение платы Arduino к сервису Wyliodrin.....	259
13.7.1. ...с помощью библиотеки Firmata	259
13.7.2. ...без использования библиотеки Firmata	266
13.8. Совместная работа Raspberry Pi и платы GrovePi.....	270
13.9. Обмен сообщениями между платами Raspberry Pi через сервис Wyliodrin.....	274
13.10. Отправка данных в сервис Wyliodrin с мобильного устройства.....	277
Глава 14. Wi-Fi модуль ESP8266	283
14.1. Режим AT-команд.....	284
14.2. Прошивка NodeMCU.....	289
14.2.1. Запуск веб-сервера	291
14.2.2. Подключение к ESP8266 модулей датчиков средствами языка Lua.....	292
14.3. Проект Home`s Smart.....	294
14.3.1. Прошивка и первоначальная настройка модуля ESP8266	296
14.3.2. Обновление прошивки через Интернет.....	300
14.3.3. Подключение датчиков к модулю ESP8266.....	300
Подключение датчика температуры DS18B20	300
Подключение датчика влажности DHT11 (DHT22).....	302
Подключение I ² C-датчика освещенности BH1750.....	302
14.4. Отправка данных с модуля ESP8266 на сайт «Народный мониторинг»	306
14.5. Отправка данных с модуля ESP8266 в сервис ThingSpeak	307
14.6. Подключение дисплея WH1602 к плате модуля ESP8266	310
14.7. Управление выводами GPIO модуля ESP8266.....	311
14.8. Работа с прерываниями модуля ESP8266.....	312
14.9. Управление каналами ШИМ модуля ESP8266	313
14.10. Планировщик задач модуля ESP8266	314
Заключение.....	315
Приложение. Описание электронного архива.....	316
Предметный указатель	317

ГЛАВА 1



Интернет вещей (вместо введения)

Internet of Things (IoT, Интернет вещей) — это концепция «умного дома», где все (или многие) бытовые приборы и системы управляются через Интернет.

Идея Интернета вещей впервые возникла еще в 1999 году у Кевина Эштона — исследователя из Массачусетского технологического института (MIT), предложившего тогда концепцию системы управления через Интернет промышленными объектами. Интернет вещей предполагает оснащение каждого устройства, будь то пылесос, холодильник или стиральная машина, модулем подключения к Интернету с возможностью взаимодействия его с домашним компьютером или смартфоном домовладельца.

С появлением Интернета вещей автоматически решится множество самых разных проблем: от индивидуального комфорта и безопасности, когда «умный» дом будет оценивать и контролировать собственное состояние, до ликвидации пробок на дорогах, когда машины сами станут договариваться со светофорами об оптимальном трафике. Холодильники смогут следить за просроченными продуктами, лекарства подскажут время приема, портфель в дождливую погоду напомнит хозяину, что тот забыл зонтик, а автомобиль сам выдержит безопасную дистанцию в потоке машин и покажет, где и как лучше припарковаться.

Интернет вещей — это не только множество различных приборов и датчиков, объединенных между собой проводными и беспроводными каналами связи и подключенных к сети Интернет, а более тесная интеграция реального и виртуального миров, в котором общение осуществляется между людьми и устройствами.

В этой книге мы познакомимся с практическими примерами создания простейших устройств для Интернета вещей на базе популярного контроллера Arduino и микрокомпьютера Raspberry Pi. Чтобы Arduino и Raspberry Pi стали полноценными устройствами для Интернета вещей, их необходимо оснастить датчиками и исполнительными устройствами и предоставить им доступ к сети Интернет. Соответственно, мы рассмотрим работу Arduino и Raspberry Pi с различными датчиками и устройствами, а также организацию доступа их к сети с дальнейшей отправкой данных в известные облачные сервисы и получением их оттуда.

А последняя, 14-я, глава книги посвящена использованию в качестве устройства для Интернета вещей нового, но быстро набирающего популярность Wi-Fi модуля ESP8266.

ГЛАВА 2



Среда программирования Arduino IDE

Разработка собственных приложений на базе плат, совместимых с архитектурой Arduino, осуществляется в официально бесплатной среде программирования Arduino IDE. Среда предназначена для написания, компиляции и загрузки собственных программ в память микроконтроллера, установленного на плате Arduino-совместимого устройства. Основой среды разработки является язык Processing/Wiring — это фактически обычный C++, дополненный простыми и понятными функциями для управления вводом/выводом на контактах устройства. Для операционных систем Windows, Mac OS и Linux существуют свои версии среды. Скачать среду Arduino IDE можно с ее официального сайта: www.arduino.cc.

Последняя версия Arduino IDE — 1.6.5 — имеет множество улучшений по сравнению с предыдущими. Вот далеко неполный их список:

- включена поддержка значительного количества платформ;
- организовано определение и отображение плат в меню списка портов вместе с последовательным портом;
- увеличена скорость компиляции;
- добавлено автосохранение при компиляции/загрузке скетча;
- в основу монитора последовательного порта положена современная библиотека JSSC (вместо старой RXTX), что дало возможность повысить его быстродействие;
- для опций **Найти/Заменить** организовано несколько вкладок;
- улучшено множество библиотек Arduino IDE (String, Serial, Print и пр.);
- обновлены инструменты и компиляторы (avr-gcc, arm-gcc, avrdude, bossac);
- переработан интерфейс командной строки;
- добавлен вывод информации о размере скетча и использовании памяти;
- в редакторе теперь отображаются номера строк;
- меню с большим количеством строк имеют полосы прокрутки;
- организована загрузка устройства Arduino Yun через сеть;

- ❑ улучшен класс `HardwareSerial`;
- ❑ увеличены стабильность и производительность USB;
- ❑ библиотека SPI теперь поддерживает транзакции для улучшения совместимости при использовании одновременно нескольких SPI-устройств;
- ❑ появилась возможность настроить подменю с конфигурациями;
- ❑ устранены проблемы загрузки на Leonardo, Micro и Yun;
- ❑ усовершенствованы библиотеки для Arduino — в частности: Bridge, TFT, Ethernet, Robot_Control, SoftwareSerial, GSM — и устранены проблемы в их работе;
- ❑ устранено множество незначительных ошибок пользовательского интерфейса.

2.1. Установка Arduino IDE

2.1.1. В ОС Windows

Отправляемся на страницу <http://arduino.cc/en/Main/Software> (рис. 2.1), выбираем версию для операционной системы Windows и скачиваем соответствующий архивный файл. Его объем составляет чуть более 80 Мбайт и содержит все необходимое, в том числе и драйверы. По окончании загрузки распаковываем скачанный файл в удобное для себя место.

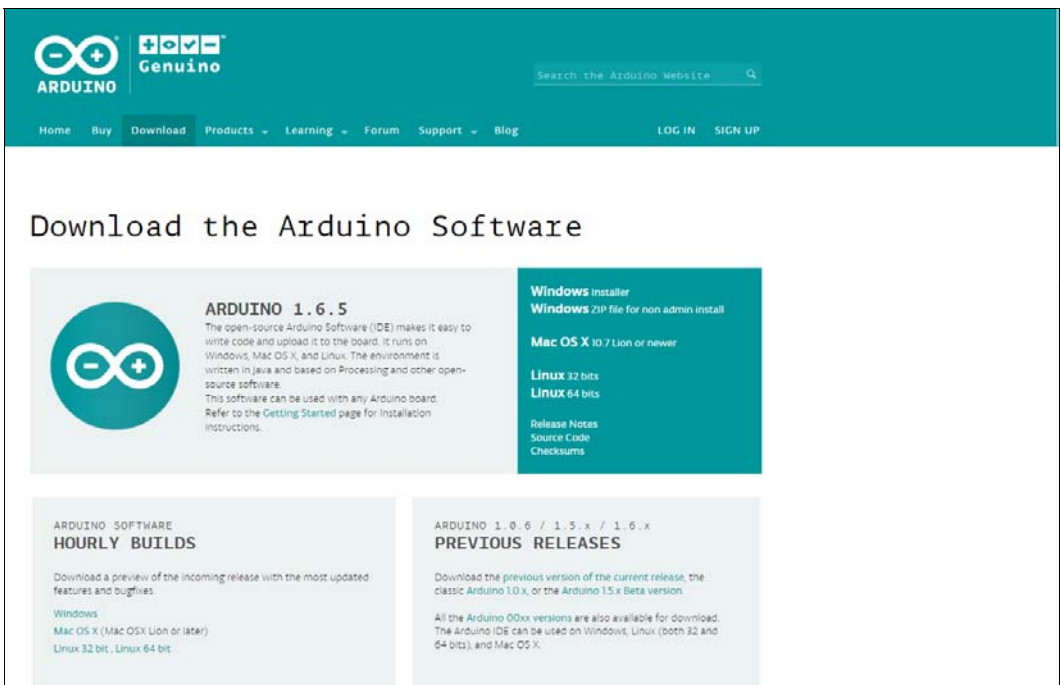


Рис. 2.1. Страница загрузки официального сайта Arduino

Для установки драйверов подключаем устройство (пусть это будет Arduino Uno) к компьютеру — на контроллере должен загореться индикатор питания (зеленый светодиод). Начавшаяся тут же попытка Windows автоматически установить драйверы заканчивается сообщением: **Программное обеспечение драйвера не было установлено.**

Не беда: открываем Диспетчер устройств и в составе устройств находим значок Arduino Uno — он там помечен восклицательным знаком. Щелкаем правой кнопкой мыши на этом значке и в открывшемся окне выбираем опцию **Обновить драйверы** и далее — **Выполнить поиск драйверов на этом компьютере.** Указываем путь к драйверам — ту папку на компьютере, куда распаковывали скачанный архив, — пусть это будет папка drivers каталога установки Arduino (например, C:\arduino-1.6.5\drivers). Игнорируем все предупреждения Windows и получаем в результате сообщение: **Обновление программного обеспечения для данного устройства завершено успешно.** В заголовке окна будет указан и COM-порт, на который установлено устройство.

Осталось запустить среду разработки Arduino IDE (рис. 2.2). Как уже отмечалось ранее, в новой версии Arduino IDE в списке доступных портов отображается и название подключенной платы Arduino.

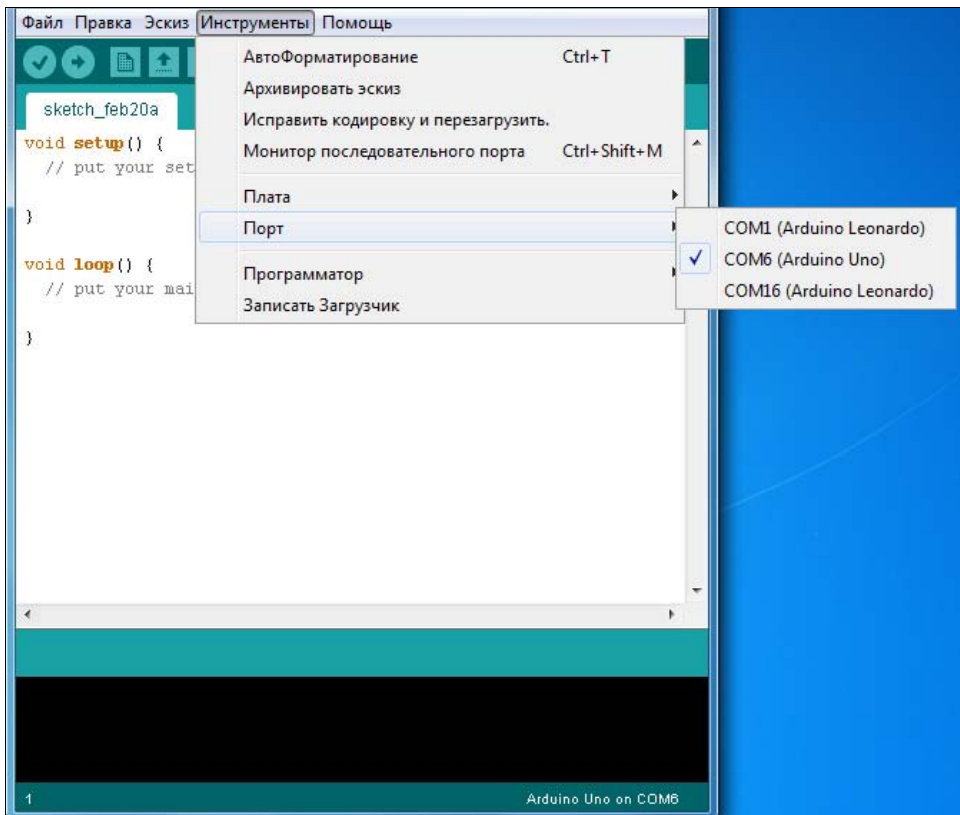


Рис. 2.2. Среда разработки Arduino IDE: выбор порта

2.1.2. В ОС Linux

В Linux Ubuntu среда Arduino IDE устанавливается еще проще, поскольку она находится в репозитории стандартных приложений Linux.

Итак, в меню Ubuntu **Приложения | Центр приложений Ubuntu | Загрузить приложение** выбираем из списка доступных программ Arduino IDE, затем в списке разделов выбираем **Инструменты разработчика**, в списке следующего уровня — **Все приложения** и в следующем открывшемся списке — **Arduino IDE**. В открывшемся окне (рис. 2.3) щелкаем левой кнопкой мыши на значке этой программы — справа от нее появляется кнопка **Установить**, нажимаем на эту кнопку, и среда устанавливается автоматически. Для запуска Arduino IDE выбираем опцию меню **Приложения | Программирование | Arduino IDE**.

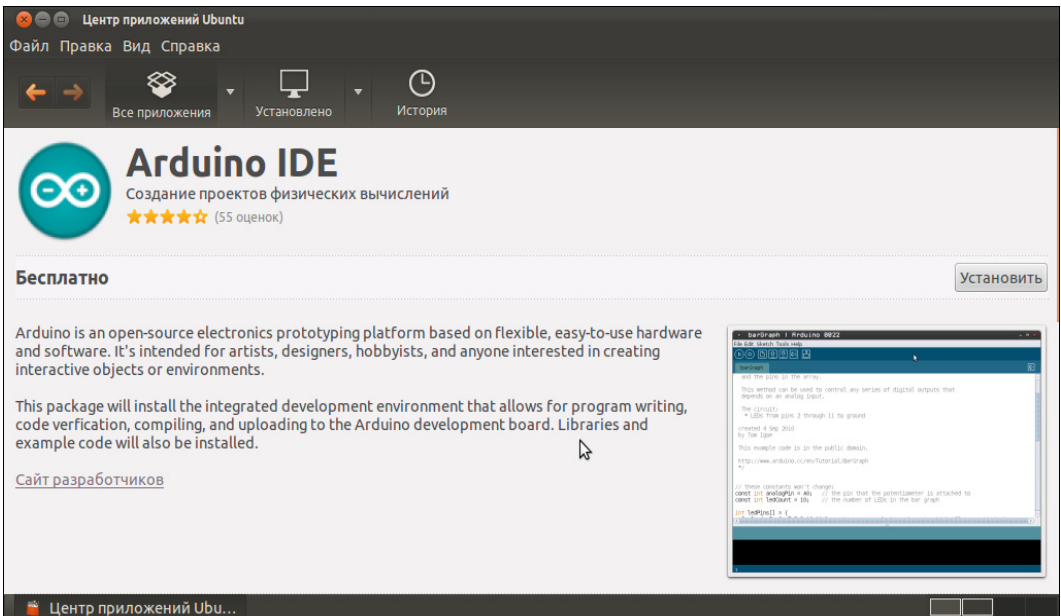


Рис. 2.3. Установка Arduino IDE из центра приложений Ubuntu

Надо заметить, что при таком способе устанавливается не последняя версия программы Arduino IDE. И чтобы работать именно с ее последней версией, нужно скачать со страницы загрузки официального сайта проекта Arduino (<https://www.arduino.cc/en/Main/Software>) архив с версией программы для Linux (см. рис. 2.1) и распаковать его в желаемое место — например, в `/home/<user>/Arduino`. Осталось для запуска программы выполнить из терминала команды:

```
cd ~/Arduino
./arduino
```

2.1.3. В Mac OS X

Для установки Arduino IDE в операционной системе Mac OS X, как и в предыдущих случаях, скачиваем со страницы загрузки официального сайта проекта Arduino (<https://www.arduino.cc/en/Main/Software>) архив с версией программы для OS X (см. рис. 2.1), распаковываем его и копируем содержимое архива в папку **Программы** — после чего значок **Arduino** появляется в списке программ **Launchpad** (рис. 2.4).



Рис. 2.4. Установка Arduino IDE в OS X

2.2. Настройка Arduino IDE

Среда разработки Arduino состоит из:

- редактора программного кода;
- области сообщений;
- окна вывода текста;
- панели инструментов с кнопками часто используемых команд;
- нескольких меню.

Программа, написанная в среде Arduino, носит название *скетч*. Скетч пишется в текстовом редакторе, который имеет цветовую подсветку создаваемого про-

граммного кода. Во время сохранения и экспорта проекта в области сообщений появляются пояснения и информация об ошибках. Окно вывода текста показывает сообщения Arduino, включающие полные отчеты об ошибках и другую информацию. Кнопки панели инструментов позволяют проверить и записать программу, создать, открыть и сохранить скетч, открыть мониторинг последовательной шины.

Дополнительная функциональность может быть добавлена разрабатываемым скетчам с помощью *библиотек*, представляющих собой специальным образом оформленный программный код, реализующий некоторый функционал, который можно подключить к создаваемому проекту. Специализированных библиотек существует множество. Обычно библиотеки пишутся так, чтобы упростить решение той или иной задачи и скрыть от разработчика детали программно-аппаратной реализации. Среда Arduino IDE поставляется с набором стандартных библиотек: Serial, EEPROM, SPI, Wire и др. Они находятся в подкаталоге *libraries* каталога установки Arduino. Внутри каталога с именем библиотеки находятся файлы *.cpp и *.h.

Необходимые библиотеки могут быть также загружены с различных ресурсов — папка загруженной библиотеки просто копируется в каталог стандартных библиотек (тот же самый подкаталог *libraries* каталога установки Arduino). Многие библиотеки снабжаются примерами, расположенными в папке *examples*. Если библиотека установлена правильно, то она появляется в меню **Эскиз | Импорт библиотек**. Выбор библиотеки в меню приведет к добавлению в исходный код строчки:

```
#include <имя библиотеки.h>
```

Эта директива подключает заголовочный файл библиотеки с описанием ее объектов, функций и констант, которые теперь могут быть использованы в проекте, поскольку среда Arduino станет компилировать создаваемый проект уже вместе с указанной библиотекой.

При загрузке скетча используется загрузчик (bootloader) Arduino — небольшая программа, загружаемая в микроконтроллер на плате. Она позволяет загружать программный код без использования дополнительных аппаратных средств. Работа загрузчика распознается по миганию светодиода на цифровом выводе D13.

Перед загрузкой скетча требуется задать необходимые параметры в меню **Инструменты | Плата** (рис. 2.5) и **Инструменты | Порт** (см. рис. 2.2).

Современные платформы Arduino перед загрузкой перезагружаются автоматически. На старых же платформах для этого необходимо нажать кнопку перезагрузки. На большинстве плат во время процесса загрузки мигают светодиоды RX и TX.

Монитор последовательного порта (Serial Monitor) отображает данные, посылаемые в платформу Arduino (плату USB или плату последовательной шины). Для отправки данных необходимо ввести в соответствующее поле текст и нажать кнопку **Отправить** (Send) или клавишу <Enter> (рис. 2.6), после чего из выпадающего списка в правом нижнем углу окна монитора выбрать скорость передачи, соответствующую значению `Serial.begin` в скетче.

На Mac OS или в Linux при подключении мониторинга последовательной шины платформа Arduino будет перезагружена (скетч начнется сначала).

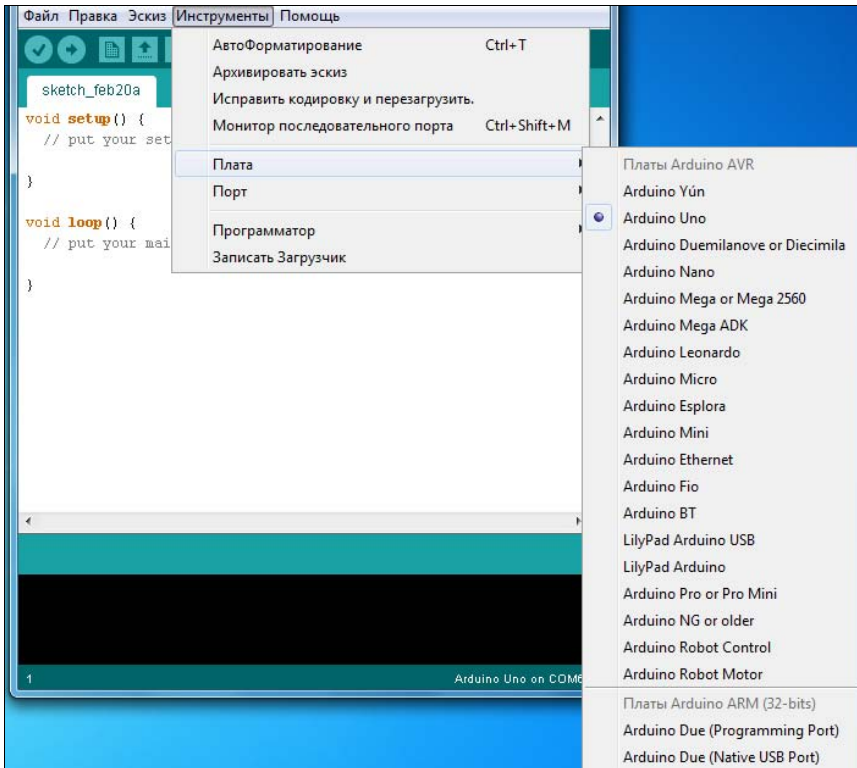


Рис. 2.5. Arduino IDE: выбор платы

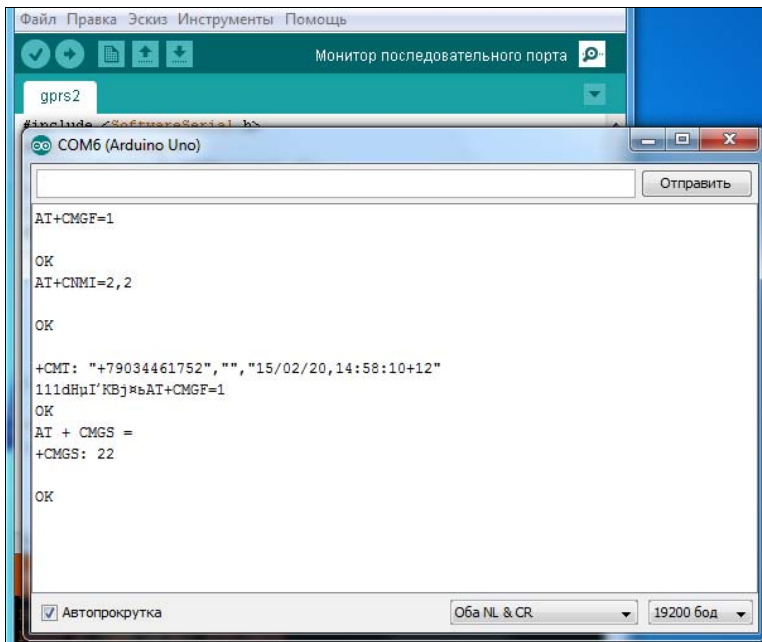


Рис. 2.6. Arduino IDE: монитор последовательного порта

ГЛАВА 3



Среда разработки Fritzing

Среда разработки Fritzing представляет собой инструмент разработчика с открытым исходным кодом, предназначенный для прототипирования и создания программного кода на базе Arduino, а также и для обучения пользователей этим процедурам. Среда работает как на Windows, так и на Mac OS и на Linux.

С помощью Fritzing можно создать принципиальную схему устройства и оформить ее представление в виде соединения макетов элементов, разработать печатную плату, написать код программы и загрузить его в плату Arduino. В отличие от других систем проектирования, у Fritzing простой интерфейс, который делает разработку электронных схем интуитивно понятной.

3.1. Загрузка и установка среды Fritzing

Для установки среды разработки Fritzing перейдите на страницу ее загрузки (<http://fritzing.org/download/?donation=0>) и выберите свою операционную систему (рис. 3.1). На момент подготовки этой книги актуальна версия среды 0.9.2b. Для установки программы на компьютер следуйте инструкциям, приведенным на странице загрузки.

3.2. Главное окно среды Fritzing

В первый раз открыв среду разработки Fritzing, вы увидите окно, изображенное на рис. 3.2.

Для разработки схемы соединения элементов разрабатываемого нами проекта перейдем на вкладку **Макетная плата** (рис. 3.3).

В правой части окна находятся панели инструментов со всеми элементами и опциями. Они содержат как основные компоненты: провода, кнопки, резисторы и пр., так и различные специализированные компоненты — например, платы Arduino и датчики.

Если компонент настраивается, то в панели инструментов **Инспектор** отображаются настраиваемые параметры для этого компонента (рис. 3.4).

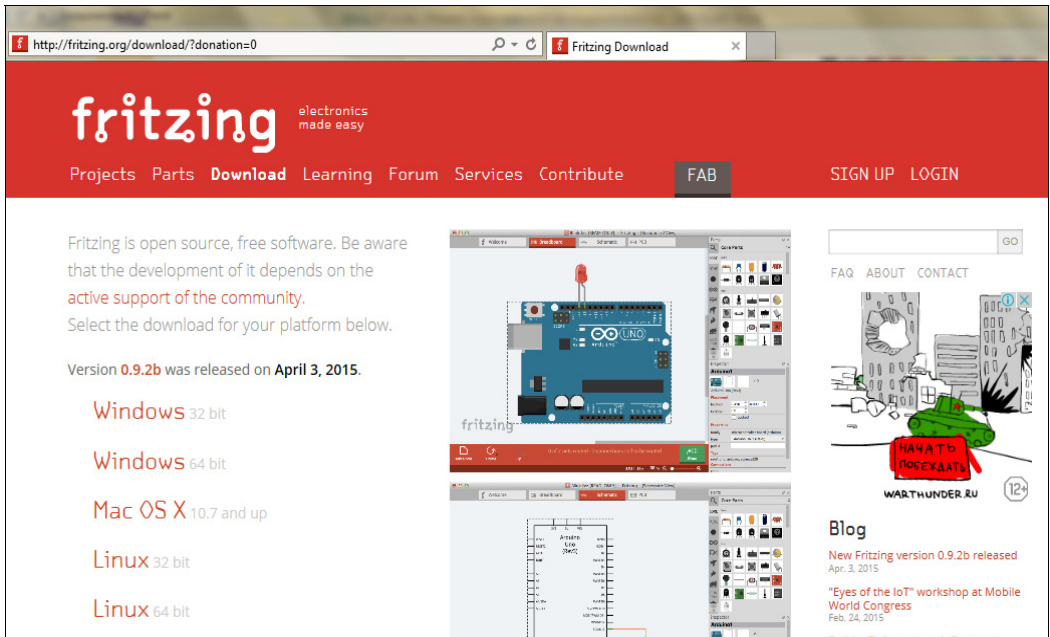


Рис. 3.1. Страница загрузки Fritzing

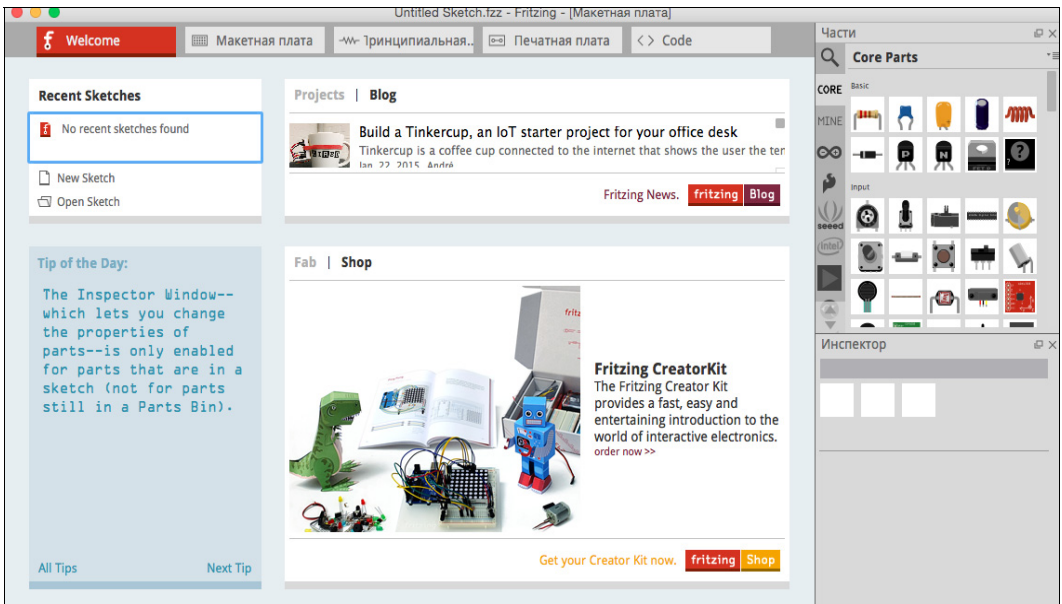


Рис. 3.2. Окно программы Fritzing

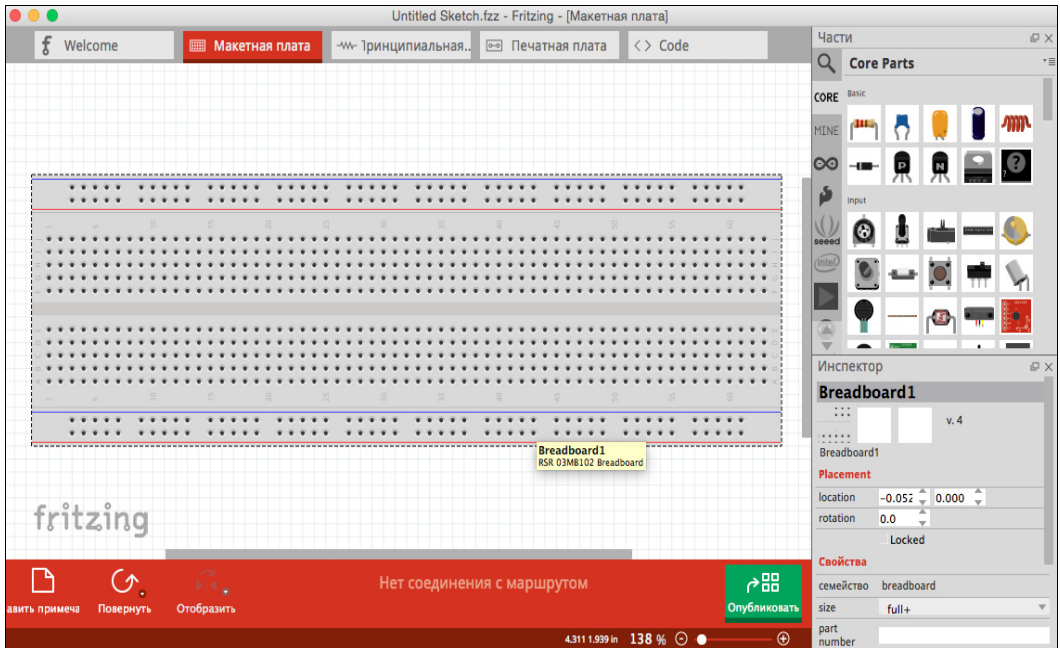


Рис. 3.3. Вкладка Макетная плата

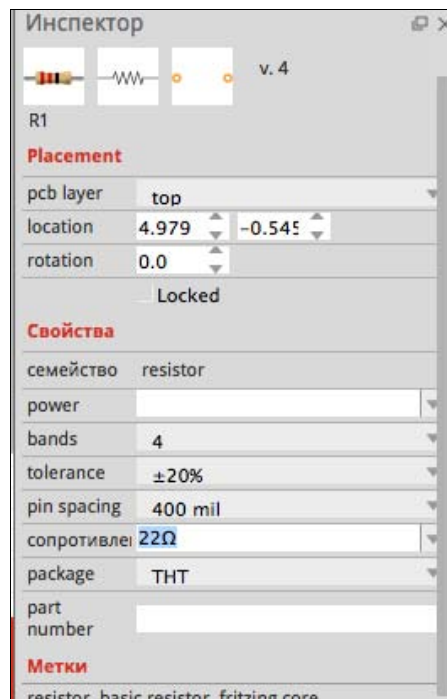


Рис. 3.4. Панель Инспектор для настройки параметров компонента

3.3. Создание схемы соединений

В качестве примера соберем небольшую схему, в которой к выводу 13 платы Arduino Uno будет подключен светодиод. Для этого добавим на вкладку **Макетная плата** следующие элементы: Arduino Uno, светодиод и резистор.

Резистор перетащим на макетную плату так, чтобы каждый его вывод попал на отдельный столбец на плате. При этом, когда компонент подключается к тому или иному отверстию макетной платы, весь столбец отверстий становится светло-зеленым, а зеленая линия указывает на электрическое соединение между отверстиями. В свойствах компонента на панели **Инспектор** выберем номинал резистора: **220 Ом**.

Затем поместим на плату светодиод. Здесь так же, как и на реальной макетной плате, мы можем добавлять провода, чтобы подключать необходимые нам элементы: для добавления провода наведите курсор мыши на отверстие на макетной плате (оно подсвечивается синим — это означает, что можно начинать вести провод), щелкните на нем левой кнопкой мыши и, не отпуская кнопки, перетащите второй конец провода в требуемую точку. Добавив таким образом необходимые провода, мы получим схему, представленную на рис. 3.5. Чтобы использовать ее в дальнейшем, необходимо сохранить ее в нужном формате (командой меню **Файл | Экспорт**).

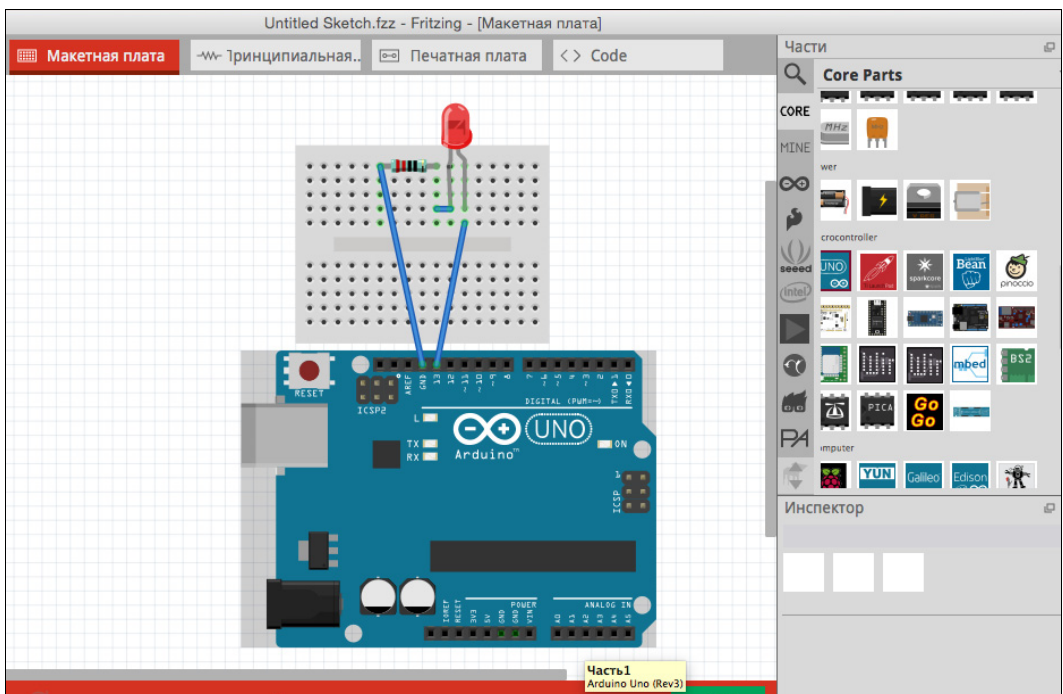


Рис. 3.5. Схема подключения светодиода к выводу 13 Arduino Uno

3.4. Создание принципиальной схемы

В предыдущем разделе мы рассмотрели создание *схемы соединений*. Теперь создадим на основе этой схемы (см. рис. 3.5) *принципиальную схему* устройства, для чего переключимся на вкладку **Принципиальная схема** (рис. 3.6).

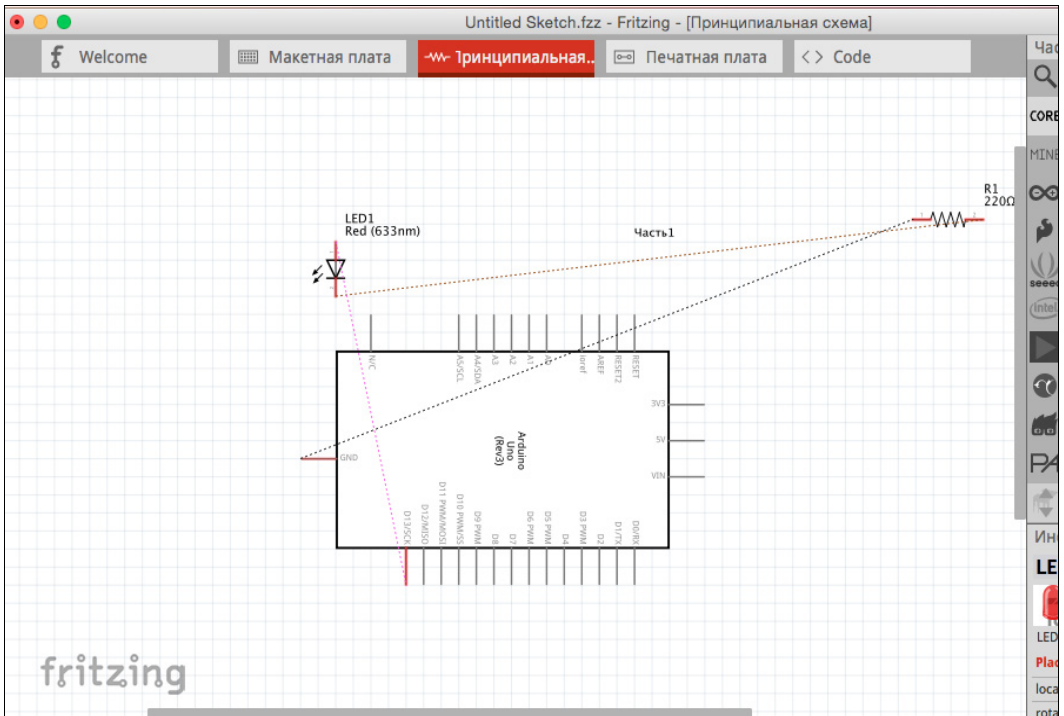


Рис. 3.6. Вкладка **Принципиальная схема** для нашей схемы соединений: исходное состояние

Как можно видеть, среда Fritzing создала все необходимые соединения, надо только привести их к более опрятному виду. Для этого в меню Fritzing выбираем команду **Вид | Подогнать окно** — чтобы автоматически сцентрировать и отмасштабировать схему на рабочем поле. Затем, перетаскивая и поворачивая компоненты, постараемся добиться того, чтобы проводники не пересекались, или же количество таких пересечений стало минимальным (рис. 3.7).

После этого в левом нижнем углу окна среды Fritzing нажимаем кнопку **Автотрассировка**, и схема приводится в порядок. Теперь ее можно сохранить для публикации в удобном формате (командой меню **Файл | Экспорт**).

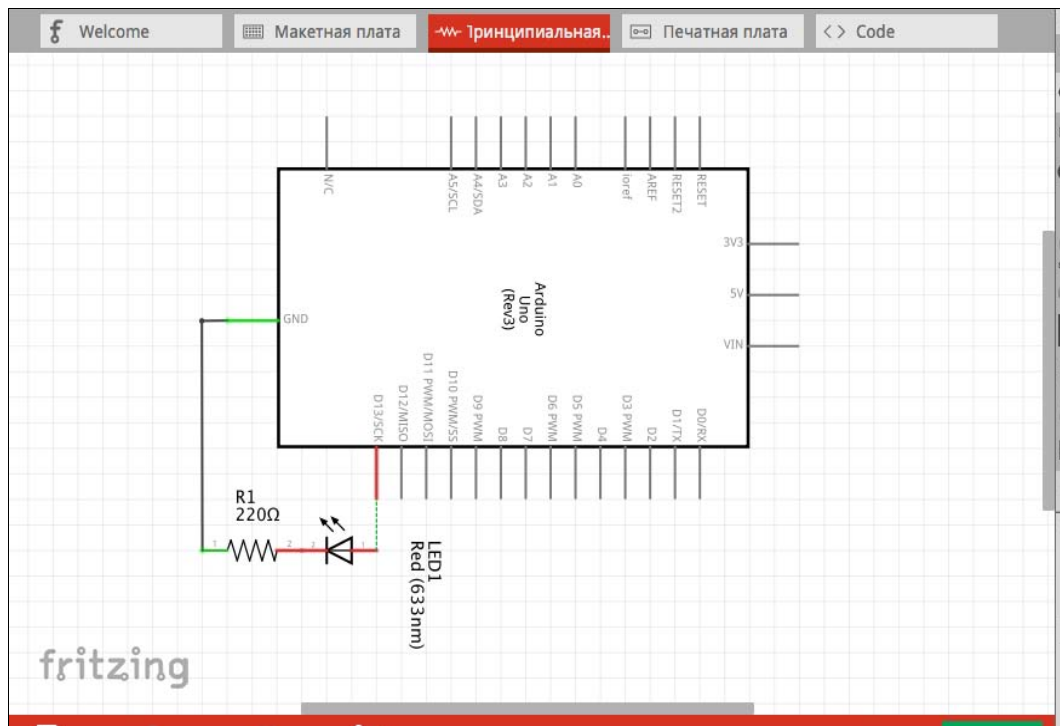


Рис. 3.7. Вкладка Принципиальная схема для нашей схемы соединений: состояние после необходимой корректировки

3.5. Добавление компонентов в среду Fritzing

Панели инструментов в правой части окна Fritzing содержат библиотеки уже имеющихся распространенных элементов, а также некоторые модули ряда производителей. Все это мы можем использовать сразу после установки Fritzing. Но рано или поздно возникает необходимость добавить какой-либо компонент или шилд, с которым мы планируем работать далее. Как же добавить новый компонент в библиотеку Fritzing?

Вы можете создать собственный компонент с нуля или найти готовые компоненты на просторах Интернета. Большое количество таких компонентов собрано на русскоязычном ресурсе «Роботоша» по адресу: <http://robotosha.ru/arduino/fritzing-library.html> (рис. 3.8). Скачаем оттуда какой-нибудь компонент — например, 16-кнопочную клавиатуру (файл KEYPAD 4x4.fzpz) и добавим ее в библиотеку Fritzing.

Итак, выбираем раздел, в который мы собираемся добавить новый компонент, — пусть это будет, например: **MINE | My Parts** (Мои компоненты). Щелкаем правой кнопкой мыши, выбираем пункт **Import** и в открывшемся окне выбираем скачан-