

Виктор Петин

Arduino и Raspberry Pi в проектах Internet of Things

2-е издание

Санкт-Петербург
«БХВ-Петербург»
2018

УДК 004.4
ББК 32.973.26-018.2
П29

Петин В. А.

П29 Arduino и Raspberry Pi в проектах Internet of Things. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2018. — 432 с.: ил. — (Электроника)

ISBN 978-5-9775-3951-7

Рассмотрено создание простых устройств в рамках концепции Интернета вещей (IoT, Internet of Things) на базе популярной платформы Arduino и микрокомпьютера Raspberry Pi. Описана установка и настройка среды разработки приложений Arduino IDE, а также среда макетирования Frizing. Раскрыты технические возможности, особенности подключения и взаимодействия различных датчиков и исполнительных устройств. Показана организация доступа устройств к сети Интернет, отправка и получение ими данных с использованием популярных облачных IoT-сервисов: Narodmon, ThingSpeak, Xively, Weaved, Blynk, Wyliodrin 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.

Во втором издании добавлены Arduino-проекты со счетчиками воды и датчиками переменного тока, проект по созданию GPS-трекера на Arduino, проекты на платформе ThingWorx для конкурсов JuniorSkills, а также усовершенствованы проекты на основе популярного Wi-Fi-модуля ESP8266 (метеостанция, отправка данных по MQTT, Wi-Fi-пульт для квадрокоптера, печать курса валют на термопринтере). 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>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</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 31.01.18.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 34,83.
Тираж 1000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

ООО "Печатное дело",
142300, МО, г. Чехов, ул. Полиграфистов, д. 1

ISBN 978-5-9775-3951-7

© ООО "БХВ", 2018
© Оформление. ООО "БХВ-Петербург", 2018

Оглавление

Глава 1. Интернет вещей (вместо введения)	9
Глава 2. Среда программирования Arduino IDE	13
2.1. Установка Arduino IDE	14
2.1.1. В ОС Windows	14
2.1.2. В ОС Linux	16
2.1.3. В Mac OS X	17
2.2. Настройка Arduino IDE	17
Глава 3. Среда разработки Fritzing	21
3.1. Загрузка и установка среды Fritzing	21
3.2. Главное окно среды Fritzing	21
3.3. Создание схемы соединений	24
3.4. Создание принципиальной схемы	25
3.5. Добавление компонентов в среду Fritzing	26
Глава 4. Arduino и аналоговые датчики	29
4.1. Аналоговые датчики (сенсоры)	29
4.2. Arduino и датчик температуры LM335	31
4.3. Arduino, Ethernet Shield/W5100 и облачные сервисы	33
4.3.1. Отправка данных на сайт «Народный мониторинг»	37
4.3.2. Чтение данных с фоторезистора	43
4.3.3. Отправка данных в сервис ThingSpeak	45
4.4. Arduino и инфракрасные датчики расстояния SHARP	53
4.4.1. Подключение датчиков Sharp к Arduino	55
4.4.2. Подсчет количества посетителей магазина	57
4.4.3. Приложение ThingTweet сервиса ThingSpeak	59
4.4.4. Отправка данных о количестве посетителей в Twitter из Arduino	61
4.5. Arduino и датчики переменного тока SCT	66
4.5.1. Подключение датчика SCT-013 Sharp к Arduino	68
4.5.2. Отправка данных в сервис ThingSpeak	71
4.6. Arduino и счетчик расхода воды	76
4.6.1. Счетчик воды Бетар СГВ-15Д	77

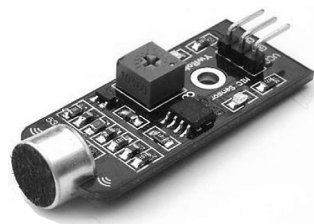
4.6.2. Подключение счетчика к плате Arduino	78
4.6.3. Отправка данных о расходе воды в «облако»	79
Глава 5. Использование Arduino в качестве контроллера исполнительных устройств	83
5.1. Arduino и электромагнитное реле	83
5.1.1. Электромагнитное реле	83
5.1.2. Устройство и принцип работы электромагнитного реле	84
5.1.3. Подключение реле к Arduino	85
5.2. Arduino и твердотельное реле	87
5.3. Arduino и диммер	88
5.3.1. Диммер	88
5.3.2. Подключение диммера к Arduino	89
5.3.3. Скетч управления диммером	90
5.4. Arduino и сервоприводы	92
5.4.1. Принципы управления сервоприводами	93
5.4.2. Управление сервоприводом с помощью Arduino	95
5.5. Arduino и библиотека TinyWebServer	97
5.5.1. Использование файлов с SD-карты для формирования веб-страниц	98
5.5.2. Включение/выключение реле с веб-страницы	99
5.5.3. Веб-страница для управления реле	100
5.5.4. Веб-страница для управления сервоприводом	105
Глава 6. Arduino и устройства I²C	109
6.1. Обзор протокола I ² C	109
6.2. Arduino и библиотека Wire	113
6.3. Arduino и датчик освещенности BH1750 на шине I ² C	116
6.4. Arduino и сервис Xively	119
6.4.1. Отправка данных в сервис Xively	122
6.4.2. Получение данных из сервиса Xively	125
6.5. Arduino и датчик влажности и температуры SHT21 на шине I ² C	127
6.6. Arduino и сервис Xively (продолжение)	130
6.6.1. Отправка мультиданных в сервис Xively	130
6.6.2. Получение мультиданных из сервиса Xively	133
6.7. Arduino и часы реального времени на шине I ² C	136
6.8. Arduino и SD-карта: чтение и запись данных	141
Глава 7. Arduino и 1-Wire	145
7.1. Технология 1-Wire	145
7.2. Применение 1-Wire	148
7.3. Интерфейс 1-Wire	149
7.3.1. Обмен информацией по шине 1-Wire	150
7.3.2. Протокол обмена информацией 1-Wire	153
7.4. Arduino и цифровой датчик температуры DS18B20	155
7.4.1. Цифровой датчик температуры DS18B20	155
7.4.2. Использование библиотеки OneWire для получения данных температуры с датчика DS18B20	158

Глава 8. Сервер для сбора данных с Ethernet-модулей датчиков, установленных на Arduino	161
8.1. Датчики влажности DHT11 и DHT22	161
8.1.1. Подключение датчиков DHT к Arduino	163
8.1.2. Библиотека DHT	163
8.2. Модуль датчика движения HC-SR501	165
8.3. Модуль датчика звука FC-04	168
8.4. Ethernet-модуль датчиков на Arduino	169
8.5. Сервер сбора данных	173
Глава 9. Обмен данными с помощью платы GPRS/GSM Shield.....	177
9.1. Отправка и получение SMS-сообщений	179
9.2. Отправка данных на сайт «Народный мониторинг»	182
9.3. GPS-трекер на Arduino и GPRS/GSM Shield	187
Глава 10. Проект Blynk: управление Arduino с планшета.....	193
10.1. Начало работы: тестовый пример	194
10.2. Управление с планшета исполнительными устройствами, подключенными к Arduino	202
10.3. Отправка данных из Arduino на экран планшета	205
Глава 11. IoT-платформа ThingWorx	211
11.1. Подключение к платформе ThingWorx	212
11.2. Мини-теплица на Arduino	212
11.3. Создание «вещи» в ThingWorx и задание ее свойств	218
11.4. Изменение свойств «вещи» в ThingWorx.....	221
11.5. Создание веб-страницы для отображения принимаемых данных	226
11.6. Отправка данных с Arduino в сервис ThingWorx	233
11.7. Создание в ThingWorx потока данных и построение графиков	239
11.8. Создание на веб-странице кнопок для отправки команд управления актуаторами на Arduino	243
11.9. Создание скетча для получения Arduino-устройством данных из ThingWorx	249
Глава 12. Микрокомпьютер Raspberry Pi.....	253
12.1. Технические характеристики и возможности Raspberry Pi.....	254
12.2. Установка операционной системы	257
12.3. Первоначальная настройка ОС Raspbian	260
12.3.1. Меню конфигурации	260
12.3.2. Настройка сетевых параметров	262
12.3.3. Настройка доступа по Wi-Fi	262
12.3.4. Подключение 3G-модема	265
12.4. Интерфейс GPIO	269
12.4.1. Управление GPIO из оболочки bash	272
12.4.2. Управление GPIO командами языка Python	272
12.5. Raspberry Pi и датчик температуры DS18B20 на шине 1-Wire	275
12.5.1. Подключение датчика DS18B20 к Raspberry Pi.....	275
12.5.2. Отправка данных с датчика DA18B20 в сервис «Народный мониторинг»	278
12.6. Raspberry Pi и датчик освещенности BH1750 на шине I ² C	281
12.6.1. Подключение датчика BH1750 к Raspberry Pi.....	281
12.6.2. Получение на Raspberry Pi данных с датчика BH1750.....	283

Глава 13. WebIOPi — веб-интерфейс и облако для Raspberry Pi	285
13.1. Установка WebIOPi на ОС Raspbian	285
13.2. Задание пользовательского пароля WebIOPi	287
13.3. Настройка сервера WebIOPi	288
13.4. JavaScript-библиотека webiop.js	289
13.4.1. Функции библиотеки webiop.js	290
13.5. Проект управления веб-камерой на сервоприводах	296
13.6. WebIOPi — подключение устройств	302
13.7. Доступ к устройству из сервиса Weaved	304
13.7.1. Установка сервиса Weaved.....	304
13.7.2. Подключение к Raspberry Pi в сервисе Weaved.....	308
Глава 14. Проект Wylidrin: управление удаленными устройствами из браузера	311
14.1. Добавление устройства в профиль	312
14.2. Запись образа Wylidrin на SD-карту.....	313
14.2.1. ...в ОС Windows	314
14.2.2. ...в ОС Linux	315
14.2.3. ...в Mac OS.....	316
14.2.4. ...в ОС Raspbian	316
14.3. Запись на SD-карту настроек Wylidrin.....	316
14.4. Подключение Raspberry Pi к Wylidrin.....	317
14.5. Создание приложения в графической среде программирования	319
14.6. Включение/выключение светодиода с веб-страницы.....	323
14.7. Подключение платы Arduino к сервису Wylidrin.....	325
14.7.1. ...с помощью библиотеки Firmata	325
14.7.2. ...без использования библиотеки Firmata	332
14.8. Совместная работа Raspberry Pi и платы GrovePi.....	336
14.9. Обмен сообщениями между платами Raspberry Pi через сервис Wylidrin.....	340
14.10. Отправка данных в сервис Wylidrin с мобильного устройства.....	343
Глава 15. Wi-Fi-модуль ESP8266	349
15.1. Режим AT-команд.....	350
15.2. Отладочные платы NodeMCU	355
15.3. Прошивка NodeMCU.....	356
15.3.1. Запуск веб-сервера	358
15.3.2. Подключение к ESP8266 модулей датчиков средствами языка Lua.....	359
15.4. Arduino IDE для ESP8266.....	361
15.5. Подключение модулей ESP8266 к сети Интернет по Wi-Fi.....	364
15.6. Метеостанция на ESP8266	366
15.6.1. Подключение датчика температуры DS18B20	367
15.6.2. Подключение датчика влажности DHT22.....	370
15.7. Отправка данных метеостанции в сервис ThingSpeak.....	372
15.8. Протокол MQTT — простой протокол для Интернета вещей.....	378
15.9. Использование MQTT-сервера CloudMQTT	379
15.10. Мобильное приложение IoT Manager	382
15.11. Отправка данных брокеру CloudMQTT для приложения IoT Manager (тестовый пример).....	384

15.12. Отправка данных метеостанции по MQTT в сервис CloudMQTT для приложения IoT Manager.....	387
15.13. Создание на модуле ESP8266 пульта для управления квадрокоптером AR.Drone 2.0	392
15.13.1. Подключение модуля ESP8266 ESP-07 к квадрокоптеру по Wi-Fi.....	392
15.13.2. Удаленное программирование квадрокоптера AR.Drone через Wi-Fi.....	392
15.13.3. Получение навигационных данных от квадрокоптера AR.Drone.....	394
15.13.4. Отправка команд взлета и посадки	400
15.13.5. Подключение гироскопа-акселерометра MPU6050 для управления AR.Drone 2.0	402
15.14. Печать курса валют на термопринтере	405
15.14.1. Подключение термопринтера к модулю NodeMCU ESP8266	405
15.14.2. Подключение модуля DS3231 к модулю NodeMCU ESP8266	408
15.14.3. Получение XML-файла с курсом валют с сайта cbr.ru.....	411
15.14.4. Обработка данных курса валют и печать на принтер.....	413
15.15. Интернет-часы на модулях ESP8266 и TM1637 с синхронизацией по NTP.....	415
Заключение.....	421
Приложение. Описание электронного архива.....	423
Предметный указатель	425

ГЛАВА 1



Интернет вещей (вместо введения)

Интернет вещей (Internet of Things, IoT) — это широкая сеть объектов, связанных через Интернет и способных обмениваться данными.

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.

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

Решения на базе Интернета вещей становятся сейчас все более востребованными именно потому, что дают поставщикам «умных» решений возможность получать дополнительную прибыль, — «умное» поведение может дать существенный прирост «полезности», потребительской стоимости устройства или системы. Так, вентилятор, который «сам» выключается при достижении нужной температуры, экономит владельцу электроэнергию и поэтому может стоить для него дороже. А вентилятор, который еще и «видит», когда в помещении есть люди, а когда нет, — ценен еще больше.

Но как техника может стать «умной»? Во-первых, за счет, собственно, своей конструкции — эта конструкция может быть такой, что поведение системы будет выглядеть разумным.

Во-вторых, за счет «интеллектуализации» — оснащения системы устройствами сбора информации, ее обработки и принятия решений. Такой подход позволяет обеспечить достаточно сложное и «разумное» поведение гораздо более простыми способами, чем за счет создания соответствующей конструкции.

Наконец, третий путь — поведение системы становится «разумным» вследствие того, что она взаимодействует с другими системами. Так, для экономии энергии

системе отопления требуется краткосрочный прогноз погоды. Этот прогноз можно получить, установив соответствующие датчики и систему обработки информации с них, способную прогнозировать погоду (мини-метеостанцию), а можно просто запросить погоду в Интернете. И в том, и в другом случае поведение системы отопления будет выглядеть «разумным».

Важно, что в последнем примере с точки зрения заказчика система ведет себя практически одинаково — соответственно, заказчик готов заплатить за эту функциональность одну и ту же цену. Однако для поставщика такой системы организация подключения ее к Интернету будет стоить значительно дешевле, чем разработка интеллектуальной метеостанции.

Благодаря интеллекту и коннективности у оборудования появляется новый набор функций. Их можно разделить на четыре группы:

- мониторинг;
- управление;
- оптимизация;
- автономность.

Каждая функция, важная и сама по себе, оказывается своего рода ступенькой для следующего уровня. Например, функция *мониторинга* служит основой для управления, оптимизации и автономности техники. Компания может выбирать такой набор функций, чтобы ее продукция была максимально полезной для потребителя, — и тем самым укреплять свою конкурентную позицию.

Возьмем, к примеру, автоматическую теплицу, которая самостоятельно осуществляет полив, поддержание нужной температуры, уровня освещенности и пр. Такая теплица окажется востребованной теми, кто не хочет тратить много времени на уход за растениями, а также может не иметь для этого возможности в периоды длительного отсутствия: командировок, отпуска и т. п.

Какую проблему клиента решит функция мониторинга? Прежде всего — устранил беспокойство насчет того, все ли в порядке с растениями во время его отсутствия: есть ли вода в системе, не выключалось ли электричество, может ли система вентиляции обеспечить нужную температуру, если в помещении стало слишком жарко, и др.

Клиент наверняка заплатит больше, если предоставить ему возможность в любой момент знать, каковы условия в его теплице.

Таким образом, продажная стоимость теплицы с функцией удаленного мониторинга параметров может возрасти существенно, в то время, как ее реализация для производителя будет достаточно простой. В результате применение технологии Интернета вещей позволит производителю получить дополнительную прибыль.

Еще выше потребительская стоимость будет у той же теплицы, если добавить функцию *управления*, — чтобы пользователь мог удаленно не только получать информацию об условиях в теплице, но и менять их по своему усмотрению.

Наверняка в теплице подогрев включается автоматически, если температура падает ниже заданного предела, но, возможно, не стоит его включать, если знать, что по

прогнозу погоды совсем скоро ожидается повышение температуры? Таким образом, функция *оптимизации* за счет использования дополнительной информации позволит сэкономить деньги на содержание теплицы и получить урожай с меньшими затратами.

Наконец, средствами Интернета вещей несложно начать следить за количеством расходуемых материалов — к примеру, удобрений, — и автоматизировать их заказ, либо контролировать состояние элементов, требующих замены или обслуживания: насосов, вентиляторов, нагревающих элементов, организовав таким образом самодиагностику и самообслуживание теплицы вплоть до полной ее *автономности*.

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, *15-я глава* книги, посвящена использованию в качестве устройства для Интернета вещей новой, но быстро набирающей популярность платформы, — модулей Wi-Fi ESP8266.

Все необходимые для работы листинги из этой книги вы найдете в сопровождающем книгу электронном архиве, который можно загрузить с FTP-сервера издательства «БХВ-Петербург» по ссылке: <ftp://ftp.bhv.ru/9785977539517.zip> или со страницы книги на сайте www.bhv.ru (см. *приложение*). Архив также содержит библиотеки Arduino и модуля ESP8266, используемые в примерах и проектах книги и не включенные в среду разработки Arduino IDE.

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.

The screenshot shows the Arduino website's software download page. At the top, there is the Arduino logo and the Genuino logo. A search bar is located in the top right corner. Below the navigation bar, the main heading is "Download the Arduino Software". The central content area is divided into two main sections. The left section is for "ARDUINO 1.6.5", featuring a large circular logo with the Arduino infinity symbol. The text describes the IDE as open-source and easy to use on Windows, Mac OS X, and Linux. It provides links for "Windows installer", "Windows ZIP file for non admin install", "Mac OS X 10.7 Lion or newer", "Linux 32 bits", and "Linux 64 bits". There are also links for "Release Notes", "Source Code", and "Checksums". The right section is for "ARUINO 1.0.6 / 1.5.x / 1.6.x PREVIOUS RELEASES", which offers a download for the previous version of the current release or the classic Arduino 1.0.x. Below these are two smaller sections: "ARUINO SOFTWARE HOURLY BUILDS" and "ARUINO 1.0.6 / 1.5.x / 1.6.x PREVIOUS RELEASES".

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.

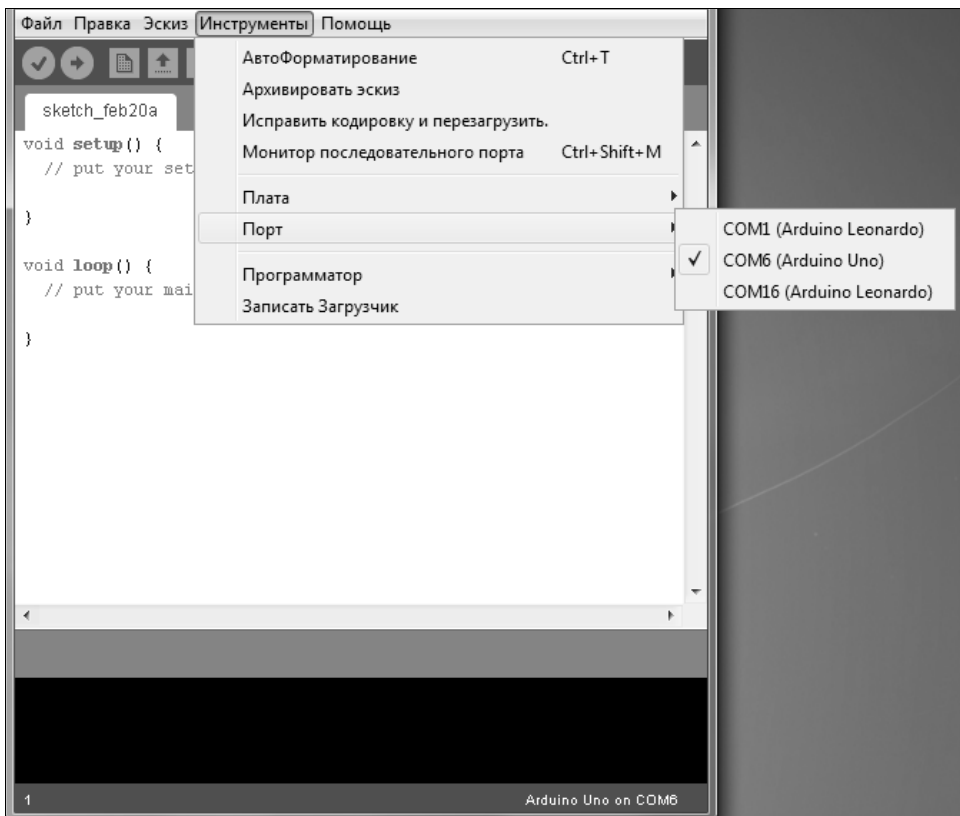


Рис. 2.2. Среда разработки Arduino IDE: выбор порта

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**.

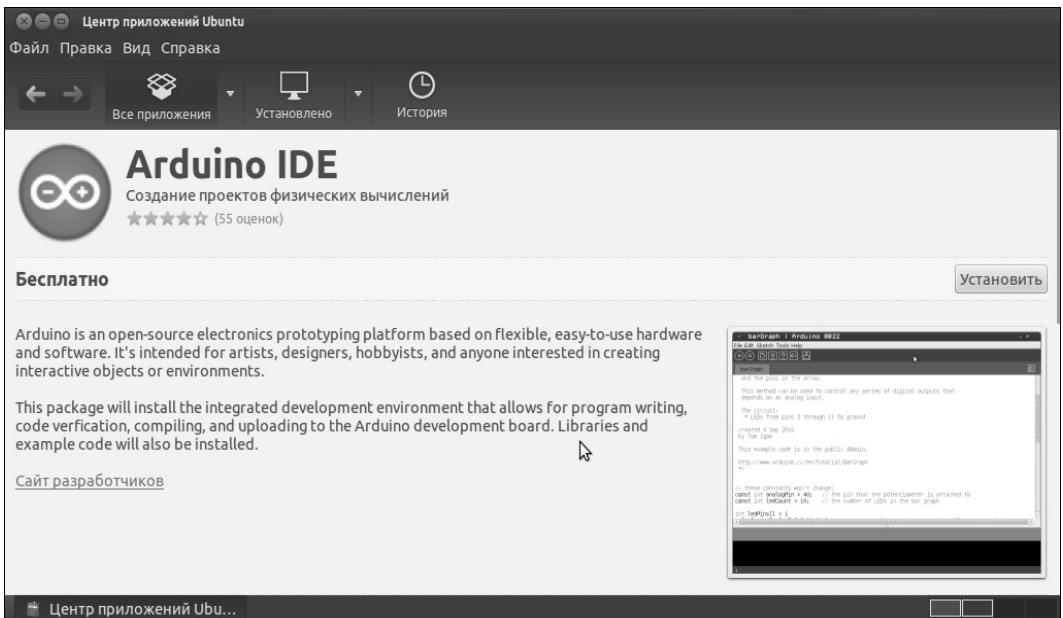


Рис. 2.3. Установка Arduino IDE из центра приложений Ubuntu

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).

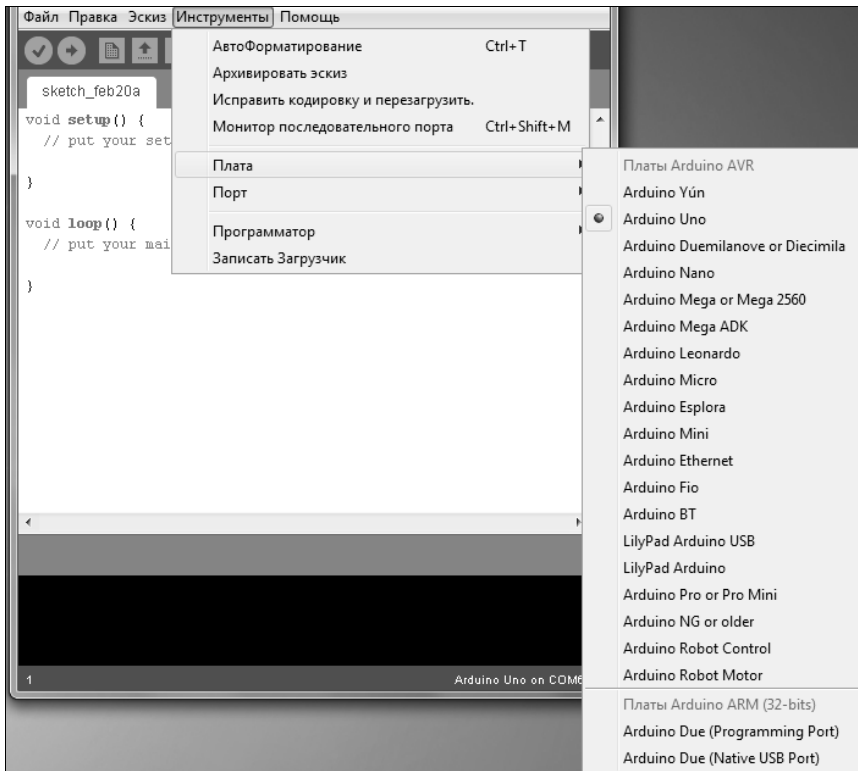


Рис. 2.5. Arduino IDE: выбор платы

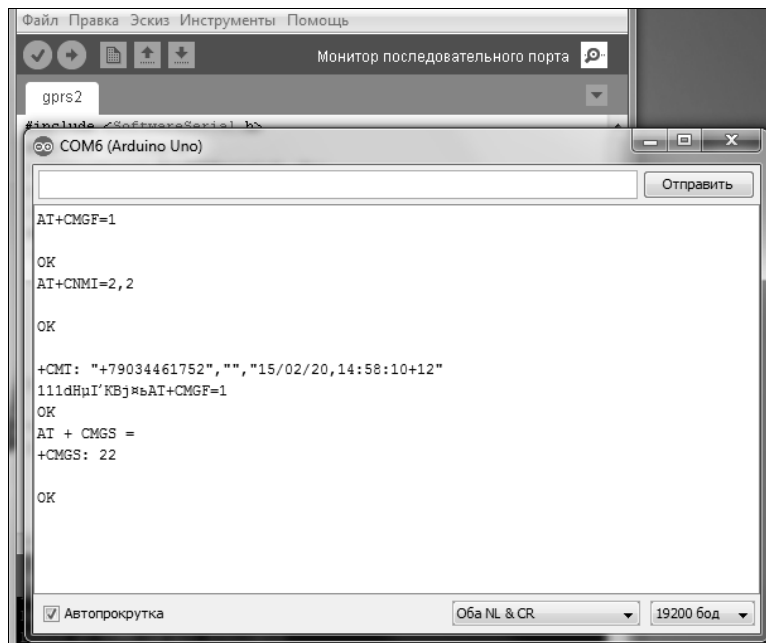


Рис. 2.6. Arduino IDE: монитор последовательного порта

