

# Электроника

Виктор Петин

ARDUINO

OPEN SOURCE ELECTRONICS  
PROTOTYPING PLATFORM

MADE IN ITALY

WWW.ARDUINO.IT



## Проекты с использованием контроллера **Arduino**



Материалы  
на [www.bhv.ru](http://www.bhv.ru)

- исходные коды программ проектов из книги
- исходные коды библиотек
- описания и спецификация электронных компонентов
- схемы из книги в формате sp17

Arduino — проекты любой  
сложности легко и быстро!

Виктор Петин

# Проекты с использованием контроллера **Arduino**

Санкт-Петербург

«БХВ-Петербург»

2014

УДК 004.4  
ББК 32.973.26-018.2  
П29

**Петин В. А.**

П29 Проекты с использованием контроллера Arduino. — СПб.: БХВ-Петербург, 2014. — 400 с.: ил. — (Электроника)

ISBN 978-5-9775-3337-9

Рассмотрены основные платы Arduino и платы расширения (шилды), добавляющие функциональность основной плате. Подробно описан язык и среда программирования Arduino IDE. Тщательно разобраны проекты с использованием контроллеров семейства Arduino. Это проекты в области робототехники, создания погодных метеостанций, "умного дома", вендинга, телевидения, Интернета, беспроводной связи (bluetooth, радиуправление). Для всех проектов представлены схемы и исходный код. Также приведен исходный код для устройств Android, используемых в проектах для связи с контроллерами Arduino. На сайте издательства размещен архив с исходными кодами программ проектов из книги, исходными кодами библиотек, описаниями и спецификациями электронных компонентов, схемами из книги в формате spl7.

*Для читателей, интересующихся современной электроникой*

УДК 004.4  
ББК 32.973.26-018.2

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Григорий Добин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 28.02.14.  
Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 32,25.  
Тираж 1700 экз. Заказ №  
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.  
Первая Академическая типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3337-9

© Петин В. А., 2014  
© Оформление, издательство "БХВ-Петербург", 2014

# Оглавление

<b>Предисловие</b> .....	<b>13</b>
Для кого и о чем эта книга? .....	13
Структура книги.....	13
Благодарности.....	14
<b>ЧАСТЬ I. ARDUINO — ОБЩИЙ ОБЗОР</b> .....	<b>15</b>
<b>Глава 1. Введение в Arduino</b> .....	<b>17</b>
1.1. Arduino — что это?.....	17
1.2. В чем преимущество Arduino? .....	18
1.3. История создания Arduino .....	18
<b>Глава 2. Обзор контроллеров семейства Arduino</b> .....	<b>20</b>
2.1. Arduino Pro Mini.....	21
2.2. Arduino Duemilanove.....	21
2.3. Arduino Nano .....	23
2.4. Arduino LilyPad .....	24
2.5. Arduino Uno .....	25
2.6. Arduino Mega2560.....	26
2.7. Arduino Leonardo.....	27
2.8. Arduino Due .....	28
<b>Глава 3. Платы расширения Arduino</b> .....	<b>29</b>
<b>ЧАСТЬ II. СРЕДА РАЗРАБОТКИ И ЯЗЫК ПРОГРАММИРОВАНИЯ КОНТРОЛЛЕРОВ ARDUINO</b> .....	<b>35</b>
<b>Глава 4. Среда программирования Arduino IDE</b> .....	<b>37</b>
4.1. Установка Arduino IDE в Windows.....	37
4.2. Установка Arduino IDE в Linux .....	39
4.3. Настройка средв Arduino IDE.....	39
<b>Глава 5. Программирование в Arduino</b> .....	<b>43</b>
5.1. Базовые знания.....	43
5.1.1. Цифровые выходы.....	43
5.1.2. Аналоговые входы .....	44
5.1.3. Широтно-импульсная модуляция .....	44
5.1.4. Память в Arduino .....	44

5.2. Структура программы .....	46
5.2.1. Функции <i>setup()</i> и <i>loop()</i> .....	46
5.3. Синтаксис и операторы .....	47
5.3.1. Управляющие операторы .....	47
5.3.1.1. Оператор <i>if</i> (условие) и операторы сравнения <i>==</i> , <i>!=</i> , <i>&lt;</i> , <i>&gt;</i> .....	47
5.3.1.2. Оператор <i>if..else</i> .....	47
5.3.1.3. Оператор <i>for</i> .....	48
5.3.1.4. Оператор <i>switch</i> .....	49
5.3.1.5. Оператор <i>while</i> .....	49
5.3.1.6. Оператор <i>do ... while</i> .....	50
5.3.1.7. Оператор <i>break</i> .....	50
5.3.1.8. Оператор <i>continue</i> .....	50
5.3.1.9. Оператор <i>return</i> .....	51
5.3.2. Синтаксис .....	51
5.3.2.1. <i>;</i> (точка с запятой) ; (semicolon) .....	51
5.3.2.2. <i>{}</i> (фигурные скобки) <i>{}</i> (curly braces) .....	51
5.3.2.3. Комментарии <i>//</i> (single line comment), <i>/* */</i> (multi-line comment) .....	52
5.3.3. Арифметические операторы .....	52
5.3.3.1. <i>=</i> (assignment) = оператор присваивания .....	52
5.3.3.2. <i>+</i> (сложение), <i>-</i> (вычитание), <i>*</i> (умножение), <i>/</i> (деление) .....	53
5.3.3.3. <i>%</i> (modulo) .....	53
5.3.4. Операторы сравнения .....	53
5.3.5. Логические операторы .....	53
5.3.5.1. <i>&amp;&amp;</i> (логическое И) .....	53
5.3.5.2. <i>  </i> (логическое ИЛИ) .....	53
5.3.5.3. <i>!</i> (логическое отрицание) .....	54
5.3.6. Унарные операторы .....	54
5.3.6.1. <i>++</i> (увеличение значения) / <i>--</i> (уменьшение значения) .....	54
5.3.6.2. <i>+=</i> , <i>-=</i> , <i>*=</i> , <i>/=</i> .....	54
5.4. Данные .....	54
5.4.1. Типы данных .....	54
5.4.1.1. <i>boolean</i> .....	55
5.4.1.2. <i>char</i> .....	55
5.4.1.3. <i>byte</i> .....	55
5.4.1.4. <i>int</i> .....	55
5.4.1.5. <i>unsigned int</i> .....	56
5.4.1.6. <i>long</i> .....	56
5.4.1.7. <i>unsigned long</i> .....	56
5.4.1.8. <i>float</i> .....	57
5.4.1.9. <i>double</i> .....	57
5.4.1.10. <i>string</i> — текстовые строки .....	57
5.4.1.11. Массивы .....	58
5.4.1.12. <i>void</i> .....	59
5.4.2. Константы .....	59
5.4.3. Переменные .....	60
5.4.3.1. Объявление переменных .....	60
5.4.3.2. Границы переменных .....	60
5.4.4. Преобразование типов данных .....	61
5.4.4.1. <i>char()</i> .....	61
5.4.4.2. <i>byte()</i> .....	61

5.4.4.3. <i>int()</i> .....	61
5.4.4.4. <i>long()</i> .....	61
5.4.4.5. <i>float()</i> .....	62
5.5. Функции.....	62
5.5.1. Цифровой ввод/вывод.....	62
5.5.1.1. Функция <i>pinMode</i> .....	62
5.5.1.2. Функция <i>digitalWrite()</i> .....	62
5.5.1.3. Функция <i>digitalRead()</i> .....	63
5.5.2. Аналоговый ввод/вывод .....	64
5.5.2.1. Функция <i>analogRead()</i> .....	64
5.5.2.2. Функция <i>analogReference()</i> .....	65
5.5.2.3. Функция <i>analogWrite()</i> .....	65
5.5.3. Дополнительные функции ввода/вывода .....	67
5.5.3.1. Функция <i>tone()</i> .....	67
5.5.3.2. Функция <i>noTone()</i> .....	67
5.5.3.3. Функция <i>shiftOut()</i> .....	67
5.5.3.4. Функция <i>pulseIn()</i> .....	69
5.5.4. Работа со временем.....	70
5.5.4.1. Функция <i>millis()</i> .....	70
5.5.4.2. Функция <i>micros()</i> .....	70
5.5.4.3. Функция <i>delay()</i> .....	71
5.5.4.4. Функция <i>delayMicroseconds()</i> .....	72
5.5.5. Математические функции .....	73
5.5.5.1. Функция <i>min(x,y)</i> .....	73
5.5.5.2. Функция <i>max(x, y)</i> .....	73
5.5.5.3. Функция <i>abs()</i> .....	73
5.5.5.4. Функция <i>constrain(x, a, b)</i> .....	74
5.5.5.5. Функция <i>map(value, fromLow, fromHigh, toLow, toHigh)</i> .....	74
5.5.5.6. Функция <i>pow(base, exponent)</i> .....	75
5.5.5.7. Функция <i>sq(x)</i> .....	75
5.5.5.8. Функция <i>sqrt(x)</i> .....	75
5.5.6. Тригонометрические функции .....	76
5.5.6.1. Функция <i>sin(rad)</i> .....	76
5.5.6.2. Функция <i>cos(rad)</i> .....	76
5.5.6.3. Функция <i>tan(rad)</i> .....	76
5.5.7. Генераторы случайных значений.....	76
5.5.7.1. Функция <i>randomSeed(seed)</i> .....	76
5.5.7.2. Функция <i>random()</i> .....	77
5.5.8. Операции с битами и байтами.....	77
5.5.8.1. Функция <i>lowByte()</i> .....	78
5.5.8.2. Функция <i>highByte()</i> .....	78
5.5.8.3. Функция <i>bitRead()</i> .....	78
5.5.8.4. Функция <i>bitWrite()</i> .....	78
5.5.8.5. Функция <i>bitSet()</i> .....	79
5.5.8.6. Функция <i>bitClear()</i> .....	79
5.5.8.7. Функция <i>bit()</i> .....	79
5.5.9. Внешние прерывания.....	79
5.5.9.1. Функция <i>attachInterrupt</i> .....	80
5.5.9.2. Функция <i>detachInterrupt</i> .....	80

<b>ЧАСТЬ III. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ ARDUINO .....</b>	<b>83</b>
<b>Глава 6. Arduino и набор функций Serial.....</b>	<b>85</b>
6.1. Функции библиотеки <i>Serial</i> .....	86
6.1.1. Функция <i>Serial.begin()</i> .....	86
6.1.2. Функция <i>Serial.end()</i> .....	86
6.1.3. Функция <i>Serial.available()</i> .....	86
6.1.4. Функция <i>Serial.read()</i> .....	87
6.1.5. Функция <i>Serial.flush()</i> .....	88
6.1.6. Функция <i>Serial.print()</i> .....	88
6.1.7. Функция <i>Serial.println()</i> .....	89
6.1.8. Функция <i>Serial.write()</i> .....	89
6.1.9. Функция <i>Serial.peek()</i> .....	89
6.2. Библиотека <i>SoftwareSerial</i> .....	90
6.2.1. Функция <i>SoftwareSerial()</i> .....	90
6.2.2. Функция <i>SoftwareSerial.listen()</i> .....	90
6.2.3. Функция <i>SoftwareSerial.isListening()</i> .....	91
6.2.4. Функция <i>SoftwareSerial.overflow()</i> .....	92
<b>Глава 7. Arduino и знаковосинтезирующие жидкокристаллические индикаторы .....</b>	<b>93</b>
7.1. Принцип работы модулей ЖКИ WINSTAR WH1604.....	94
7.2. Библиотека <i>LiquidCrystal</i> .....	99
7.2.1. Функция <i>LiquidCrystal()</i> .....	100
7.2.2. Функция <i>begin()</i> .....	101
7.2.3. Функция <i>clear()</i> .....	101
7.2.4. Функция <i>home()</i> .....	101
7.2.5. Функция <i>setCursor()</i> .....	101
7.2.6. Функция <i>write()</i> .....	102
7.2.7. Функция <i>print()</i> .....	102
7.2.8. Функция <i>cursor()</i> .....	103
7.2.9. Функция <i>noCursor()</i> .....	103
7.2.10. Функция <i>blink()</i> .....	103
7.2.11. Функция <i>noBlink()</i> .....	103
7.2.12. Функция <i>display()</i> .....	103
7.2.13. Функция <i>noDisplay()</i> .....	104
7.2.14. Функция <i>scrollDisplayLeft()</i> .....	104
7.2.15. Функция <i>scrollDisplayRight()</i> .....	104
7.2.16. Функция <i>autoscroll()</i> .....	104
7.2.17. Функция <i>noAutoscroll()</i> .....	104
7.2.18. Функция <i>leftToRight()</i> .....	105
7.2.19. Функция <i>rightToLeft()</i> .....	105
7.2.20. Функция <i>createChar()</i> .....	105
7.3. Библиотека <i>LiquidCrystalRus</i> .....	107
<b>Глава 8. Библиотека EEPROM.....</b>	<b>112</b>
8.1. Функции библиотек <i>EEPROM</i> .....	112
8.1.1. Функция чтения <i>EEPROM.read</i> .....	112
8.1.2. Функция записи <i>EEPROM.write</i> .....	113

8.2. Примеры использования памяти EEPROM .....	114
8.2.1. Воспроизведение звука .....	114
8.2.2. Звонок с мелодиями .....	117
<b>Глава 9. Подключение клавиатуры и мыши.....</b>	<b>122</b>
9.1. Обмен данными по протоколу PS/2 .....	122
9.2. Библиотека <i>ps2dev</i> .....	124
9.3. Подключение клавиатуры.....	125
9.4. Редактор текста на дисплее WH1604.....	126
9.5. Подключение мыши .....	134
9.6. Опрос состояния мыши.....	135
<b>Глава 10. Arduino и сенсорная панель .....</b>	<b>139</b>
10.1. Как работает резистивный экран?.....	140
10.2. Программа чтения координат сенсорного экрана .....	142
10.3. Библиотека <i>Arduino TouchScreen</i> .....	143
<b>Глава 11. Arduino и 1-Wire .....</b>	<b>144</b>
11.1. Что такое 1-Wire? .....	144
11.2. Применение 1-Wire.....	147
11.3. Протокол 1-Wire .....	148
11.3.1. Описание интерфейса 1-Wire .....	148
11.3.2. Обмен информацией по 1-Wire .....	149
11.3.3. Протокол обмена информацией 1-Wire.....	152
11.4. Библиотека <i>OneWire</i> .....	154
11.4.1. Функция <i>begin()</i> .....	155
11.4.2. Функция <i>requestFrom()</i> .....	155
11.4.3. Функция <i>beginTransmission()</i> .....	155
11.4.4. Функция <i>endTransmission()</i> .....	156
11.4.5. Функция <i>write()</i> .....	156
11.4.6. Функция <i>available()</i> .....	156
11.4.7. Функция <i>read()</i> .....	157
11.4.8. Функция <i>onReceive()</i> .....	157
11.4.9. Функция <i>onRequest()</i> .....	157
11.5. Устройство <i>iButton</i> и программирование электронного замка.....	157
11.5.1. Поиск устройств 1-Wire и получение уникального кода .....	159
11.5.2. Режимы работы.....	161
11.5.3. Режим пользователя.....	162
11.5.4. Первоначальный ввод ключа в пустую базу .....	162
11.5.5. Просмотр, запись и удаление кодов ключей в режиме администратора.....	164
11.5.6. Блок-реле. Открывание замка.....	167
11.5.7. Проигрывание мелодий .....	168
<b>Глава 12. Arduino и цифровой датчик температуры DS18B20 .....</b>	<b>170</b>
12.1. Описание датчика DS18B20 .....	170
12.2. Использование библиотеки <i>OneWire</i> .....	173
12.3. Библиотека <i>DallasTemperature</i> .....	174
<b>Глава 13. Arduino и датчики температуры и влажности DHT .....</b>	<b>176</b>
13.1. Характеристики датчиков DH11, DH22.....	176

13.2. Подключение к Arduino .....	177
13.3. Библиотека <i>DHT</i> .....	178
<b>Глава 14. Сетевой обмен с помощью Arduino .....</b>	<b>180</b>
14.1. Устройство Arduino Ethernet shield .....	180
14.2. Библиотека <i>Ethernet library</i> .....	182
14.2.1. Класс Ethernet ( <i>Ethernet class</i> ) .....	182
Функция <i>Ethernet.begin()</i> .....	183
Функция <i>Ethernet.localIP()</i> .....	183
14.2.2. Класс IPAddress ( <i>IPAddress class</i> ) .....	183
Функция <i>IPAddress()</i> .....	183
14.2.3. Класс Server ( <i>Server class</i> ) .....	184
Функция <i>ethernetServer()</i> .....	184
Функция <i>begin()</i> .....	184
Функция <i>available()</i> .....	184
Функция <i>write()</i> .....	185
Функция <i>print()</i> .....	186
Функция <i>println()</i> .....	186
14.2.4. Класс Client ( <i>Client class</i> ) .....	186
Функция <i>client()</i> .....	187
Функция <i>EthernetClient()</i> .....	187
Функция <i>connected()</i> .....	187
Функция <i>connect()</i> .....	188
Функция <i>write()</i> .....	188
Функция <i>print()</i> .....	188
Функция <i>println()</i> .....	188
Функция <i>available()</i> .....	189
Функция <i>read()</i> .....	189
Функция <i>flush()</i> .....	189
Функция <i>stop()</i> .....	190
14.2.5. Класс EthernetUDP ( <i>EthernetUDP class</i> ) .....	190
Функция <i>begin()</i> .....	190
Функция <i>read()</i> .....	190
Функция <i>write()</i> .....	191
Функция <i>beginPacket()</i> .....	191
Функция <i>endPacket()</i> .....	192
Функция <i>parsePacket()</i> .....	192
Функция <i>available()</i> .....	193
Функция <i>remoteIP()</i> .....	193
Функция <i>remotePort()</i> .....	193
14.3. Домашняя метеостанция с доступом через Интернет .....	193
14.3.1. Устройство, настройка и отладка метеостанции .....	193
14.3.2. Создание виджета для планшетов с ОС Android .....	198
14.3.3. Размещение данных метеостанции на сайте .....	199
<b>Глава 15. Arduino и карта памяти SD .....</b>	<b>201</b>
15.1. Arduino-библиотека <i>SD</i> .....	201
15.1.1. Класс <i>SD</i> .....	201
Функция <i>begin()</i> .....	202
Функция <i>exists()</i> .....	202

Функция <i>mkdir()</i> .....	202
Функция <i>rmdir()</i> .....	202
Функция <i>open()</i> .....	202
Функция <i>remove()</i> .....	203
15.1.2. Класс <i>File</i> .....	203
Функция <i>available()</i> .....	203
Функция <i>close()</i> .....	204
Функция <i>flush()</i> .....	204
Функция <i>peek()</i> .....	204
Функция <i>position()</i> .....	205
Функция <i>print()</i> .....	205
Функция <i>println()</i> .....	205
Функция <i>seek()</i> .....	206
Функция <i>size()</i> .....	206
Функция <i>read()</i> .....	206
Функция <i>write()</i> .....	206
Функция <i>isDirectory()</i> .....	207
Функция <i>openNextFile()</i> .....	207
Функция <i>rewindDirectory()</i> .....	207
15.2. Запись показаний датчиков на SD-карту.....	207
<b>Глава 16. Arduino и светодиодные матрицы.....</b>	<b>213</b>
16.1. Светодиоды и светодиодные матрицы.....	213
16.2. Светодиодная матрица FYM-23881BUG-11.....	214
16.3. SPI-расширитель выходов 74HC595.....	215
16.4. Игра "Тетрис" на светодиодных матрицах FYM-23881BUG-11.....	217
16.4.1. Управление изображением на светодиодных матрицах.....	219
16.4.2. Фигуры игры "Тетрис".....	220
16.4.3. Управление фигурами игры "Тетрис".....	222
16.4.4. Проверка столкновения фигур.....	224
16.5. Светодиодная матрица RGB.....	226
16.6. RGB-ночник, управляемый с помощью движения рук.....	227
<b>Глава 17. Работа Arduino с купюроприемником.....</b>	<b>232</b>
17.1. Купюроприемник ICT серий A7 и V7.....	232
17.2. Подключение купюроприемника ICT V7 к Arduino.....	236
17.3. Скетч для получения номинала принимаемой купюры.....	238
<b>Глава 18. Arduino и радиочастотная идентификация (RFID).....</b>	<b>240</b>
18.1. Радиочастотная идентификация.....	240
18.2. Датчик считывания RFID-карт.....	243
18.3. Скетч для считывания RFID-карт.....	245
<b>Глава 19. Arduino и датчики расстояния.....</b>	<b>248</b>
19.1. Ультразвуковые дальномеры HC-SR04.....	248
19.2. Принцип работы ультразвукового дальномера HC-SR04.....	249
19.3. Библиотека <i>Ultrasonic</i> .....	249
19.4. Инфракрасные датчики расстояния Sharp.....	250
19.5. Подключение датчиков Sharp к Arduino.....	253

<b>Глава 20. Arduino и передача данных в инфракрасном диапазоне.....</b>	<b>254</b>
20.1. Обмен данными в инфракрасном диапазоне.....	254
20.2. Протоколы для ИК-пультов.....	256
20.2.1. Протокол RC5.....	256
20.2.2. Протокол NEC.....	257
20.2.3. Протокол JVC.....	259
20.2.4. Протокол Sony.....	260
20.3. Подключение ИК-приемника.....	261
20.4. Библиотека <i>IRremote</i> .....	261
20.5. Скетч для получения кодов ИК-пульта.....	263
<b>Глава 21. Создаем робота.....</b>	<b>269</b>
21.1. Ходовая часть.....	269
21.2. Драйвер двигателей L293D.....	272
21.3. Массив возможных состояний моторов.....	275
21.4. Разработка скетча движений робота.....	277
21.5. Движение робота по линии в автономном режиме.....	280
<b>Глава 22. Arduino и шаговые двигатели.....</b>	<b>289</b>
22.1. Управление шаговым двигателем.....	290
22.2. Arduino-библиотека <i>Stepper</i> .....	291
22.2.1. Функция <i>Stepper()</i> .....	291
22.2.2. Функция <i>setSpeed(rpm)</i> .....	291
22.2.3. Функция <i>step(steps)</i> .....	292
22.3. Пример использования библиотеки <i>Stepper</i> .....	292
22.4. Arduino-библиотека <i>AccelStepper</i> .....	294
<b>Глава 23. Arduino и сервоприводы.....</b>	<b>295</b>
23.1. Сервоприводы.....	295
23.2. Arduino-библиотека <i>Servo</i> для управления сервоприводом.....	298
23.2.1. Функция <i>attach()</i> .....	299
23.2.2. Функция <i>detach()</i> .....	299
23.2.3. Функция <i>write(int)</i> .....	300
23.2.4. Функция <i>writeMicroseconds(int)</i> .....	300
23.2.5. Функция <i>read()</i> .....	300
23.2.6. Функция <i>attached()</i> .....	300
23.3. Робот-паук на сервоприводах.....	301
23.4. Скетч для управления роботом-пауком.....	305
<b>Глава 24. Arduino и Bluetooth.....</b>	<b>310</b>
24.1. "Голубой зуб".....	310
24.2. Модуль Bluetooth HC-05.....	310
24.3. Управление роботом с Android-устройства по Bluetooth.....	316
<b>Глава 25. TV-выход на Arduino.....</b>	<b>325</b>
25.1. Схема подключения.....	325
25.2. Arduino-библиотека <i>TVOut</i> .....	325
25.2.1. Функция установки режима <i>begin()</i> .....	326
25.2.2. Функции задержки.....	326

Функция <i>delay()</i> .....	326
Функция <i>delay_frame()</i> .....	326
25.2.3. Функции получения параметров .....	327
Функция <i>hres()</i> .....	327
Функция <i>vres()</i> .....	327
Функция <i>char_line()</i> .....	327
25.2.4. Основные графические функции .....	327
Функция <i>set_pixel()</i> .....	327
Функция <i>get_pixel()</i> .....	328
Функция <i>fill()</i> .....	328
Функция <i>clear_screen()</i> .....	328
Функция <i>invert()</i> .....	329
Функция <i>shift_direction()</i> .....	329
Функция <i>draw_line()</i> .....	329
Функция <i>draw_row()</i> .....	329
Функция <i>draw_column()</i> .....	330
Функция <i>draw_rect()</i> .....	330
Функция <i>draw_circle()</i> .....	331
Функция <i>bitmap()</i> .....	331
25.2.5. Функции вывода текстовой информации .....	331
Функция <i>select_font()</i> .....	332
Функция <i>print_char()</i> .....	332
Функция <i>set_cursor()</i> .....	332
Функция <i>print()</i> .....	332
Функция <i>println()</i> .....	333
25.2.6. Функции вывода аудио .....	333
Функция <i>tone()</i> .....	333
Функция <i>noTone()</i> .....	333
25.3. Создание пользовательских шрифтов .....	334
25.4. Создание графических примитивов .....	335
25.5. Создание простейшей игровой консоли .....	338
25.6. Разработка игры для игровой консоли .....	340
25.6.1. Создание переменных игры .....	340
25.6.2. Управление положением "игрока" с помощью джойстика .....	341
25.6.3. Генерация и перемещение объектов-цифр .....	343
25.6.4. Проверка столкновения "игрока" и объектов-цифр .....	344
25.6.5. Счетчик баллов "игрока" .....	345
25.6.6. Переход на новый уровень .....	346
25.6.7. Отображение данных игры на табло .....	346
25.6.8. Звуковое сопровождение игры .....	347
25.6.9. Основной цикл игры .....	347
25.6.10. Добавляем меню для выбора игр .....	348
<b>Глава 26. Arduino и радиоуправление .....</b>	<b>350</b>
26.1. Принципы формирования радиосигнала .....	351
26.2. Установка связи приемника с передатчиком .....	353
26.3. Разработка скетча приема команд для Arduino .....	354

<b>Глава 27. Arduino и беспроводной радиомодуль NRF24L01.....</b>	<b>357</b>
27.1. Радиомодуль NRF24L01.....	357
27.2. Библиотека для работы с модулем NRF24L01 .....	358
27.2.1. Параметры библиотеки <i>Mirf</i> .....	359
27.2.2. Функции библиотеки <i>Mirf</i> .....	359
Функция <i>init()</i> .....	359
Функция <i>setRADDR()</i> .....	359
Функция <i>setTADDR()</i> .....	359
Функция <i>config()</i> .....	359
Функция <i>dataReady()</i> .....	360
Функция <i>getData()</i> .....	360
Функция <i>send()</i> .....	360
Функция <i>isSending()</i> .....	360
27.3. Пример соединения двух плат Arduino с помощью модуля NRF24L01 .....	361
<b>Глава 28. Работа Arduino с USB-устройствами .....</b>	<b>364</b>
28.1. Интерфейс USB.....	364
28.2. USB Host Shield.....	365
28.3. HID-устройства USB .....	366
28.4. Подключение HID-мыши USB .....	369
28.5. Использование HID-устройства (руль Defender) для управления роботом .....	369
28.6. Подключение к Arduino Android-устройства через USB Host Shield .....	379
<b>Глава 29. Arduino и ROS.....</b>	<b>380</b>
29.1. Что такое ROS? .....	380
29.2. Дистрибутивы ROS.....	381
29.3. Установка ROS.....	381
29.4. Узлы и темы в ROS.....	382
29.5. Пакет <i>rosserial</i> .....	383
29.6. Подготовка сообщения (publisher) на Arduino .....	384
29.7. Создание подписки (subscriber) на Arduino .....	387
29.8. Связь через ROS двух плат Arduino .....	389
<b>Глава 30. Arduino и "умный дом" X10.....</b>	<b>392</b>
30.1. Система домашней автоматизации X10 .....	392
30.2. Двусторонний интерфейс TW523 .....	394
30.3. Arduino-библиотека <i>X10</i> .....	395
30.3.1. Функция <i>begin()</i> .....	395
30.3.2. Функция <i>write()</i> .....	395
30.4. Блок на Arduino для голосового управления приборами X10 .....	397
<b>Приложение 1. Список использованных источников.....</b>	<b>399</b>
<b>Приложение 2. Описание электронного архива.....</b>	<b>400</b>

# Предисловие

## Для кого и о чем эта книга?

Предлагаемая книга ориентирована на читателей, желающих быстро войти в темы программирования микроконтроллеров и использования микроконтроллеров для связи с внешними системами в проектах автоматизации и робототехники.

Книга содержит описание языка программирования плат Arduino в среде Arduino IDE и предлагает изучение предмета на реальных проектах, имеющих практическое значение. В ней вы найдете множество примеров и проектов использования Arduino, представляющих собой законченные решения, пригодные для использования в ваших проектах.

Книга сопровождается электронным архивом, содержащим исходный код всех рассмотренных примеров и проектов, а также используемые в проектах необходимые библиотеки (*см. приложение 2*). Этот электронный архив можно скачать с FTP-сервера издательства "БХВ-Петербург" по ссылке <ftp://ftp.bhv.ru/9785977533379.zip>, а также со страницы книги на сайте [www.bhv.ru](http://www.bhv.ru).

## Структура книги

Книга состоит из трех частей и включает предисловие, тридцать глав и два приложения.

*Часть I* содержит описание Arduino, обзор контроллеров семейства Arduino и плат расширения для Arduino.

В *части II* книги рассмотрены среда разработки и язык программирования для контроллеров Arduino.

*Часть III* посвящена созданию конкретных устройств на основе контроллера Arduino. Проекты содержат электрические схемы и листинги программ. Рассмотрено использование плат расширения (шилдов). В книге широко используются библиотеки Arduino.

В *приложениях* приведены перечень использованной литературы и интернет-ресурсов и описание электронного архива, сопровождающего книгу.

## Благодарности

Хочу поблагодарить родных и близких, которые с пониманием относились к потраченному на книгу (за счет общения с ними) времени.

Большая благодарность издательству "БХВ-Петербург", где поверили в необходимость этой книги, и всем сотрудникам издательства, которые помогли мне в ее создании.

Благодарю также всех читателей, купивших эту книгу, — надеюсь, она поможет им в разработке собственных проектов на основе Arduino.

---

# ЧАСТЬ I

---



## Arduino — общий обзор

**Глава 1.** Введение в Arduino

**Глава 2.** Обзор контроллеров семейства Arduino

**Глава 3.** Платы расширения Arduino





# ГЛАВА 1

## Введение в Arduino

### 1.1. Arduino — что это?

Появление первых микроконтроллеров ознаменовало начало новой эры в развитии микропроцессорной техники. Наличие в одном корпусе большинства системных устройств сделало микроконтроллер подобным обычному компьютеру. В отечественной литературе они даже назывались однокристальными микроЭВМ. Соответственно и желание использовать микроконтроллеры как обычные компьютеры появилось практически с их появлением. Но желание это сдерживалось многими факторами. Например, чтобы собрать устройство на микроконтроллере, необходимо знать основы схемотехники, устройство и работу конкретного процессора, уметь программировать на ассемблере и изготавливать электронную технику. Потребуются также программаторы, отладчики и другие вспомогательные устройства. В итоге без огромного объема знаний и дорогостоящего оборудования не обойтись. Такая ситуация долго не позволяла многим любителям использовать микроконтроллеры в своих проектах. Сейчас, с появлением устройств, дающих возможность работать с микроконтроллерами без наличия серьезной материальной базы и знания многих предметов, все изменилось. Примером такого устройства может служить проект Arduino итальянских разработчиков.

Arduino и его клоны представляют собой наборы, состоящие из готового электронного блока и программного обеспечения. Электронный блок здесь — это печатная плата с установленным микроконтроллером и минимумом элементов, необходимых для его работы. Фактически электронный блок Arduino является аналогом материнской платы современного компьютера. На нем имеются разъемы для подключения внешних устройств, а также разъем для связи с компьютером, по которому и осуществляется программирование микроконтроллера. Особенности используемых микроконтроллеров ATmega фирмы Atmel позволяют производить программирование без применения специальных программаторов. Все, что нужно для создания нового электронного устройства, — это плата Arduino, кабель связи и компьютер. Второй частью проекта Arduino является программное обеспечение для создания управляющих программ. Оно объединило в себе простейшую среду разработки и язык программирования, представляющий собой вариант языка C/C++ для микро-

контроллеров. В него добавлены элементы, позволяющие создавать программы без изучения аппаратной части. Так что для работы с Arduino практически достаточно знания только основ программирования на C/C++. Создано для Arduino и множество библиотек, содержащих код, работающий с различными устройствами.

## 1.2. В чем преимущество Arduino?

Пользователь современного компьютера не задумывается о функционировании отдельных частей ПК. Он просто запускает нужные программы и работает с ними. Точно так же и Arduino позволяет пользователю сосредоточиться на разработке проектов, а не на изучении устройства и принципов функционирования отдельных элементов. Нет надобности и в создании законченных плат и модулей. Разработчик может использовать готовые платы расширения или просто напрямую подключить к Arduino необходимые элементы. Все остальные усилия будут направлены на разработку и отладку управляющей программы на языке высокого уровня. В итоге доступ к разработке микропроцессорных устройств получили не только профессионалы, но и просто любители что-то сделать своими руками. Наличие готовых модулей и библиотек программ позволяет непрофессионалам в электронике создавать готовые работающие устройства для решения своих задач. А варианты использования Arduino ограничены только возможностями микроконтроллера и имеющегося варианта платы, ну и, конечно, фантазией разработчика.

## 1.3. История создания Arduino

В 2002 году программист Массимо Банци (Massimo Banzi) был принят на работу в должности доцента в Институт проектирования взаимодействий города Ивреа (Interaction Design Institute Ivrea, IDII) для продвижения новых способов разработки интерактивных проектов. Однако крошечный бюджет и ограниченное время доступа к лабораторной базе сводили его усилия практически на нет. В проектах Банци использовал устройство BASIC Stamp, разработанное калифорнийской компанией Parallax. Stamp представлял собой небольшую печатную плату с размещенными на ней источником питания, микроконтроллером, памятью и портами ввода/вывода для соединения с различной аппаратурой. Программирование микроконтроллера осуществлялось на языке BASIC. BASIC Stamp имел две проблемы: недостаток вычислительной мощности и достаточно высокую цену — плата с основными компонентами стоила около 100 долларов. И команда Банци решила самостоятельно создать плату, которая удовлетворяла бы всем их потребностям.

Банци и его сотрудники поставили себе целью создать устройство, представляющее собой простую, открытую и легкодоступную платформу для разработки, с ценой — не более 30 долларов — приемлемой для студенческого кармана. Хотели они и выделить чем-то свое устройство на фоне прочих. Поэтому в противовес другим производителям, экономящим на количестве выводов печатной платы, они решили добавить их как можно больше, а также сделали свою плату синей, в отличие от обычных зеленых плат.

Продукт, который создала команда, состоял из дешевых и доступных компонентов — например, базировался он на микроконтроллере ATmega328. Но главная задача состояла в том, чтобы гарантировать работу устройства по принципу *plug-and-play*, — чтобы пользователь, достав плату из коробки и подключив к компьютеру, мог немедленно приступить к работе.

Первый прототип платы был сделан в 2005 году, она имела простейший дизайн и еще не называлась *Arduino*. Чуть позже Массимо Банци придумал назвать ее так — по имени принадлежащего ему бара, расположенного в городе Ивреа. Бренд "*Arduino*" без какой-либо рекламы и привлечения средств маркетинга быстро приобрел высокую популярность в Интернете. С начала распространения продано более 250 тыс. комплектов *Arduino*, и это не учитывая множества клонов. В мире насчитывается более двухсот дистрибьюторов продукции *Arduino* — от крупных фирм, таких как *SparkFun Electronics*, до мелких компаний, работающих на местный рынок. На сегодня платформа *Arduino* представлена не одной платой, а целым их семейством. В дополнение к оригинальному проекту, называемому *Arduino Uno*, новые модели, имеющие на плате более мощные средства, носят название *Arduino Mega*, компактные модели — *Arduino Nano*, платы в водонепроницаемом корпусе — *LilyPad Arduino*, а новая плата с 32-разрядным процессором *Cortex-M3 ARM* — *Arduino Due*.

Своим успехом проект *Arduino* обязан существовавшему до него языку *Processing* и платформе *Wiring*. От этих проектов *Arduino* унаследовал одну сильную черту — удобную для пользователя среду разработки. До появления *Arduino* программирование микроконтроллеров требовало сложного и рутинного предварительного обучения. А с *Arduino* даже те, кто не имеет опыта работы с электронными устройствами, теперь могут приобщиться к миру электроники. Начинающим уже не приходится тратить много времени на изучение сопутствующего материала — они могут быстро разработать прототип, который будет полноценно рабочим.

По словам Массимо Банци, пятьдесят лет назад, чтобы написать программное обеспечение требовалась команда людей в белых халатах, которые знали все об электронных лампах. Теперь же, с появлением *Arduino*, множество людей получили возможность создавать электронные устройства самостоятельно. Как утверждает один из членов команды Банци, инженер по телекоммуникациям Дэвид Куартильез, философия *Arduino* как раз и состоит в том, что желающий разобраться в электронике может сразу же приступить к ее изучению, вместо того чтобы сначала учиться алгебре.

## ГЛАВА 2



# Обзор контроллеров семейства Arduino

Основные версии плат Arduino представлены следующими моделями:

- ❑ **Due** — плата на базе 32-битного ARM микропроцессора Cortex-M3 ARM SAM3U4E;
- ❑ **Leonardo** — плата на микроконтроллере ATmega32U4;
- ❑ **Uno** — самая популярная версия базовой платформы Arduino;
- ❑ **Duemilanove** — плата на микроконтроллере ATmega168 или ATmega328;
- ❑ **Diecimila** — версия базовой платформы Arduino USB;
- ❑ **Nano** — компактная платформа, используемая как макет. Nano подключается к компьютеру при помощи кабеля USB Mini-B;
- ❑ **Mega ADK** — версия платы Mega 2560 с поддержкой интерфейса USB-host для связи с телефонами на Android и другими устройствами с интерфейсом USB;
- ❑ **Mega2560** — плата на базе микроконтроллера ATmega2560 с использованием чипа ATmega8U2 для последовательного соединения по USB-порту;
- ❑ **Mega** — версия серии Mega на базе микроконтроллера ATmega1280;
- ❑ **Arduino BT** — платформа с модулем Bluetooth для беспроводной связи и программирования;
- ❑ **LilyPad** — платформа, разработанная для переноски, может зашиваться в ткань;
- ❑ **Fio** — платформа разработана для беспроводных применений. Fio содержит разъем для радио XBee, разъем для батареи LiPo и встроенную схему подзарядки;
- ❑ **Mini** — самая маленькая платформа Arduino;
- ❑ **Pro** — платформа, разработанная для опытных пользователей, может являться частью большего проекта;
- ❑ **Pro Mini** — как и платформа Pro, разработана для опытных пользователей, которым требуется низкая цена, меньшие размеры и дополнительная функциональность.

Рассмотрим более подробно некоторые из этих плат.

## 2.1. Arduino Pro Mini

Arduino Pro Mini (рис. 2.1) построена на микроконтроллере ATmega168.

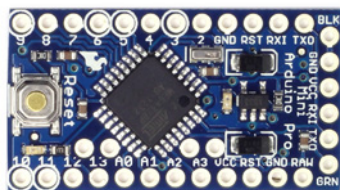


Рис. 2.1. Плата Arduino Pro Mini

Характеристики платы Arduino Pro Mini представлены в табл. 2.1.

*Таблица 2.1. Характеристики платы Arduino Pro Mini*

Микроконтроллер	ATmega168
Рабочее напряжение	3,3 В или 5 В (в зависимости от модели)
Входное напряжение	3,35–12 В (модель 3,3 В) или 5–12 В (модель 5 В)
Цифровые входы/выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Флеш-память	16 Кбайт (2 — используются для загрузчика)
ОЗУ	1 Кбайт
EEPROM	512 байтов
Тактовая частота	8 МГц (модель 3,3 В) или 16 МГц (модель 5 В)

Arduino Pro Mini может получать питание: через кабель FTDI, или от платы-конвертера, или от регулируемого источника питания 3,3 В или 5 В (зависит от модели платформы) через вывод VCC, или от нерегулируемого источника через вывод RAW.

Выводы питания:

- RAW — для подключения нерегулируемого напряжения;
- VCC — для подключения регулируемых 3,3 В или 5 В;
- GND — выводы заземления.

## 2.2. Arduino Duemilanove

Arduino Duemilanove (рис. 2.2) построена на одном из микроконтроллеров: ATmega168 или ATmega328.

Характеристики платы Arduino Duemilanove представлены в табл. 2.2.

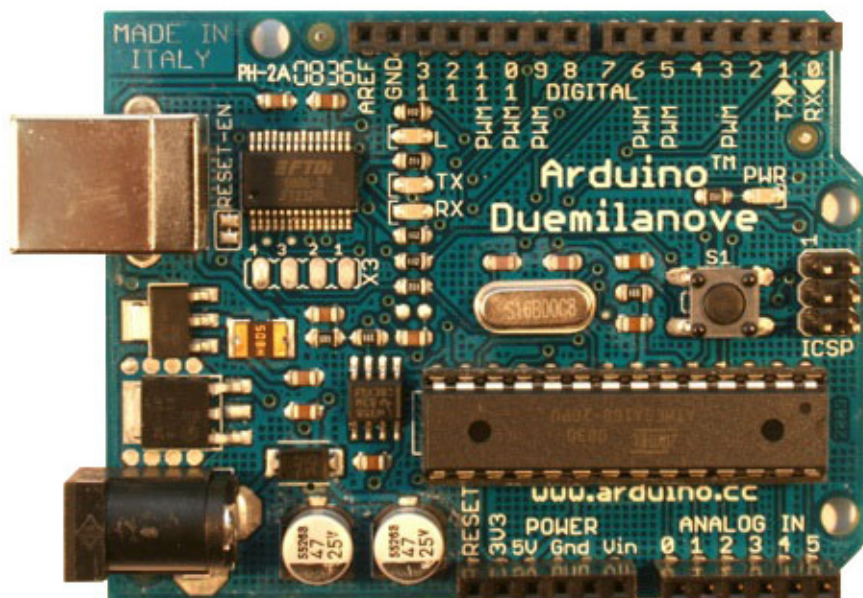


Рис. 2.2. Плата Arduino Duemilanove

Таблица 2.2. Характеристики платы Arduino Duemilanove

<b>Микроконтроллер</b>	ATmega168 или ATmega328
<b>Рабочее напряжение</b>	5 В
<b>Входное напряжение (рекомендуемое)</b>	7–12 В
<b>Входное напряжение (предельное)</b>	6–20 В
<b>Цифровые входы/выходы</b>	14 (6 из которых могут использоваться как выходы ШИМ)
<b>Аналоговые входы</b>	6
<b>Постоянный ток через вход/выход</b>	40 мА
<b>Постоянный ток для вывода 3,3 В</b>	50 мА
<b>Флеш-память</b>	16 Кбайт (ATmega168) или 32 Кбайт (ATmega328), при этом 2 Кбайт используются для загрузчика
<b>ОЗУ</b>	1 Кбайт (ATmega168) или 2 Кбайт (ATmega328)
<b>EEPROM</b>	512 байтов (ATmega168) или 1 Кбайт (ATmega328)
<b>Тактовая частота</b>	16 МГц

## 2.3. Arduino Nano

Платформа Nano (рис. 2.3), построенная на микроконтроллере ATmega328 (Arduino Nano 3.0) или ATmega168 (Arduino Nano 2.x), имеет небольшие размеры и может использоваться в лабораторных работах.



Рис. 2.3. Плата Arduino Nano

Arduino Nano может получать питание через подключение USB Mini-B, или от нерегулируемого 6–20 В (вывод 30) или регулируемого 5 В (вывод 27), внешнего источника питания. Автоматически выбирается источник с самым высоким напряжением.

Характеристики платы Arduino Nano представлены в табл. 2.3.

Таблица 2.3. Характеристики платы Arduino Nano

Микроконтроллер	ATmega168 или ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7–12 В
Входное напряжение (предельное)	6–20 В
Цифровые входы/выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3,3 В	50 мА
Флеш-память	16 Кбайт (ATmega168) или 32 Кбайт (ATmega328), при этом 2 Кбайт используются для загрузчика
ОЗУ	1 Кбайт (ATmega168) или 2 Кбайт (ATmega328)

Таблица 2.3 (окончание)

<b>EEPROM</b>	512 байтов (ATmega168) или 1 Кбайт (ATmega328)
<b>Тактовая частота</b>	16 МГц

## 2.4. Arduino LilyPad

Платформа Arduino LilyPad (рис. 2.4) разработана с целью использования в качестве части одежды. Она может быть зашита в ткань со встроенными источниками питания, датчиками и приводами с проводкой. Платформа построена на микроконтроллере ATmega168V.

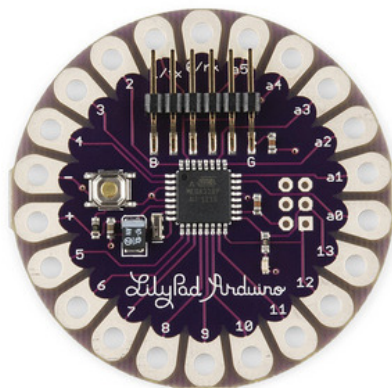


Рис. 2.4. Плата Arduino LilyPad

Характеристики платы Arduino LilyPad представлены в табл. 2.4.

Таблица 2.4. Характеристики платы Arduino LilyPad

<b>Микроконтроллер</b>	ATmega168 или ATmega328
<b>Рабочее напряжение</b>	2,7–5,5 В
<b>Входное напряжение</b>	2,7–5,5 В
<b>Цифровые входы/выходы</b>	14 (6 из которых могут использоваться как выходы ШИМ)
<b>Аналоговые входы</b>	6
<b>Постоянный ток через вход/выход</b>	40 мА
<b>Флеш-память</b>	16 Кбайт (ATmega168) или 32 Кбайт (ATmega328), при этом 2 Кбайт используются для загрузчика
<b>ОЗУ</b>	1 Кбайт (ATmega168) или 2 Кбайт (ATmega328)
<b>EEPROM</b>	512 байтов (ATmega168) или 1 Кбайт (ATmega328)
<b>Тактовая частота</b>	16 МГц

## 2.5. Arduino Uno

Контроллер Arduino Uno (рис. 2.5) построен на микроконтроллере ATmega328. В отличие от всех предыдущих плат, использовавших для связи по USB микроконтроллер FTDI USB, новый Arduino Uno использует микроконтроллер ATmega8U2.



Рис. 2.5. Плата Arduino Uno

Характеристики платы Arduino Uno представлены в табл. 2.5.

Таблица 2.5. Характеристики платы Arduino Uno

Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7–12 В
Входное напряжение (предельное)	6–20 В
Цифровые входы/выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3,3 В	50 мА
Флеш-память	32 Кбайт, при этом 0,5 Кбайт используются для загрузчика
ОЗУ	2 Кбайт
EEPROM	1 Кбайт
Тактовая частота	16 МГц

## 2.6. Arduino Mega2560

Arduino Mega (рис. 2.6) построена на микроконтроллере ATmega2560.

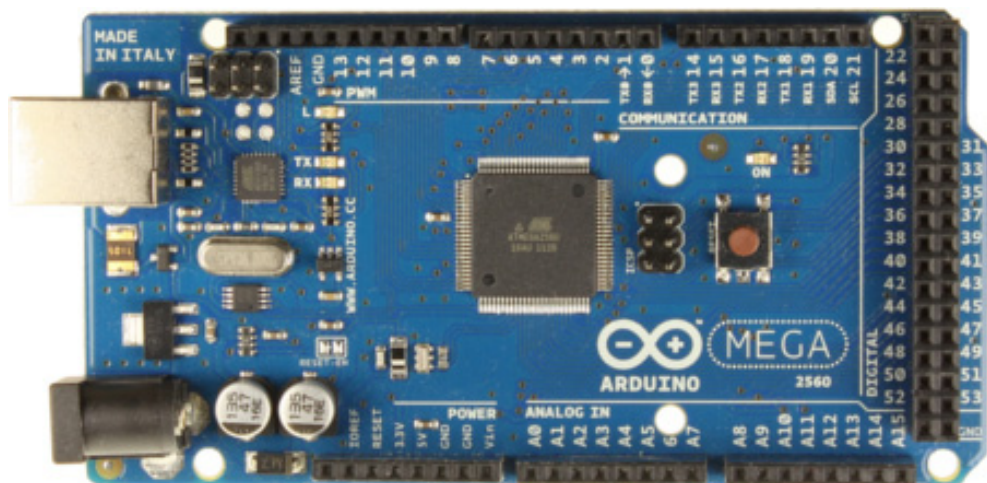


Рис. 2.6. Плата Arduino Mega2560

Характеристики платы Arduino Mega2560 представлены в табл. 2.6.

Таблица 2.6. Характеристики платы Arduino Mega2560

<b>Микроконтроллер</b>	ATmega2560
<b>Рабочее напряжение</b>	5 В
<b>Входное напряжение (рекомендуемое)</b>	7–12 В
<b>Входное напряжение (предельное)</b>	6–20 В
<b>Цифровые входы/выходы</b>	54 (14 из которых могут работать так же, как выходы ШИМ)
<b>Аналоговые входы</b>	16
<b>Постоянный ток через вход/выход</b>	40 мА
<b>Постоянный ток для вывода 3,3 В</b>	50 мА
<b>Флеш-память</b>	256 Кбайт, из которых 8 Кбайт используются для загрузчика
<b>ОЗУ</b>	8 Кбайт
<b>EEPROM</b>	4 Кбайт
<b>Тактовая частота</b>	16 МГц

## 2.7. Arduino Leonardo

Arduino Leonardo (рис. 2.7) — контроллер на базе микроконтроллера ATmega32U4. В отличие от всех предыдущих плат ATmega32U4 имеет встроенную поддержку для USB-соединения.

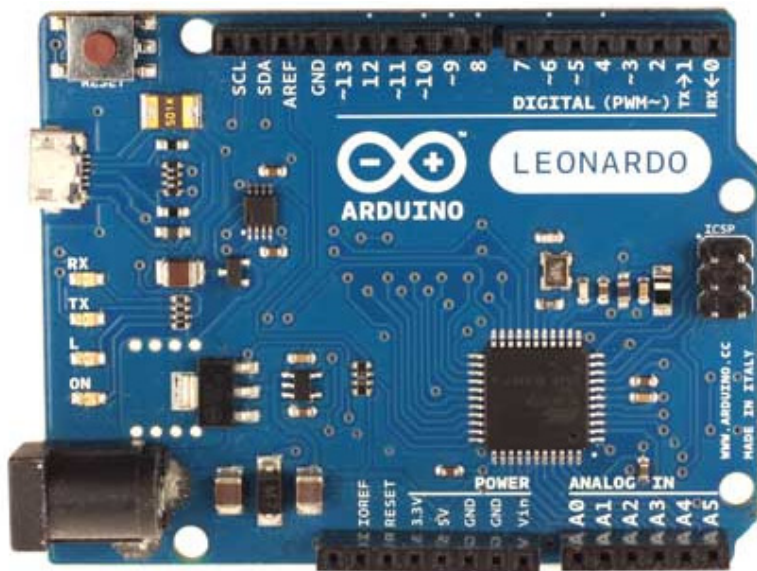


Рис. 2.7. Плата Arduino Leonardo

Характеристики платы Arduino Leonardo представлены в табл. 2.7.

Таблица 2.7. Характеристики платы Arduino Leonardo

Микроконтроллер	ATmega32U4
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7–12 В
Входное напряжение (предельное)	6–20 В
Цифровые входы/выходы	20 (7 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	12
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3,3 В	50 мА
Флеш-память	32 Кбайт, из которых 4 Кбайт используются для загрузчика
ОЗУ	2 Кбайт
EEPROM	1 Кбайт
Тактовая частота	16 МГц

## 2.8. Arduino Due

Arduino Due (рис. 2.8) — плата микроконтроллера на базе процессора Atmel SAM3X8E ARM Cortex-M3. Это первая плата Arduino на основе 32-битного микроконтроллера с ARM-ядром.

В отличие от других плат Arduino, Arduino Due работает от 3,3 В. Максимальное напряжение, которое выдерживают входы/выходы, составляет 3,3 В.

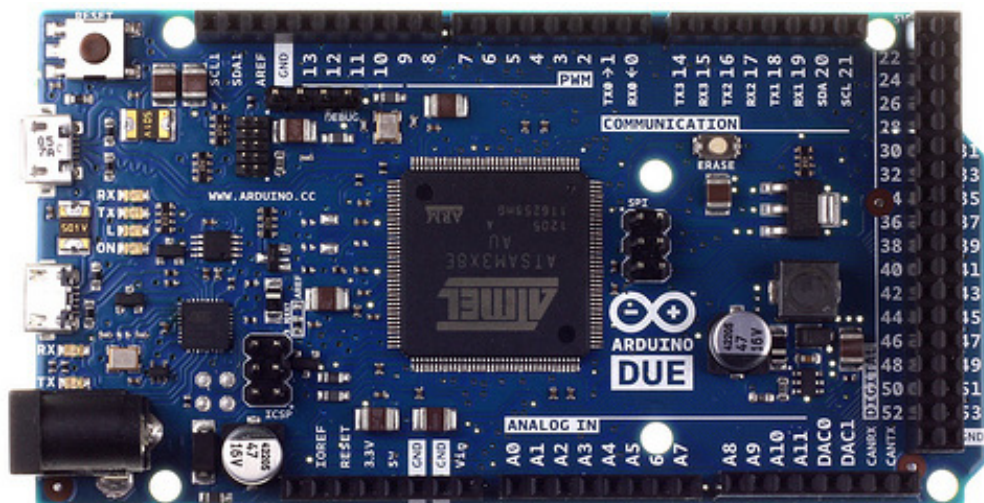


Рис. 2.8. Плата Arduino Due

Характеристики платы Arduino Due представлены в табл. 2.8.

Таблица 2.8. Характеристики платы Arduino Due

Микроконтроллер	AT91SAM3X8E
Рабочее напряжение	3,3 В
Входное напряжение (рекомендуемое)	7–12 В
Входное напряжение (предельное)	6–20 В
Цифровые входы/выходы	54 (на 12 из которых реализуется выход ШИМ)
Аналоговые входы	12
Аналоговые выходы	2 (ЦАП)
Постоянный ток через вход/выход	50 мА
Постоянный ток для вывода 3,3 В	800 мА
Постоянный ток для вывода 5 В	800 мА
Флеш-память	512 Кбайт
ОЗУ	96 Кбайт (два банка: 64 Кбайт и 32 Кбайт)
Тактовая частота	84 МГц

## ГЛАВА 3



# Платы расширения Arduino

Большую популярность плата Arduino приобрела не только из-за низкой стоимости, легкости разработки и программирования, но, главным образом, благодаря наличию плат расширения (так называемых *шилдов*), добавляющих Arduino дополнительную функциональность. Шилды (кроме маленьких модулей и платы LilyPad) подключаются к Arduino с помощью имеющихся на них штыревых разъемов (рис. 3.1).

Существует множество различных по функциональности шилдов — от простейших, предназначенных для макетирования, до сложных, представляющих собой отдельные многофункциональные устройства.

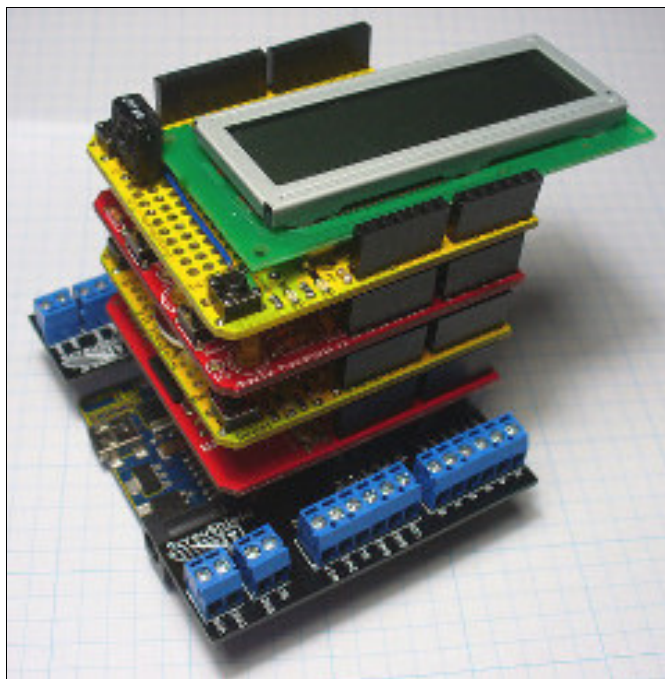


Рис. 3.1. Модульная структура установки плат расширения для Arduino

Рассмотрим примеры некоторых шилдов:

- **Ethernet Shield** (рис. 3.2) — обеспечивает подключение к Интернету;
- **XBee Shield** (рис. 3.3) — обеспечивает при помощи модуля Maxstream Xbee Zigbee беспроводную связь несколькими устройствам Arduino;

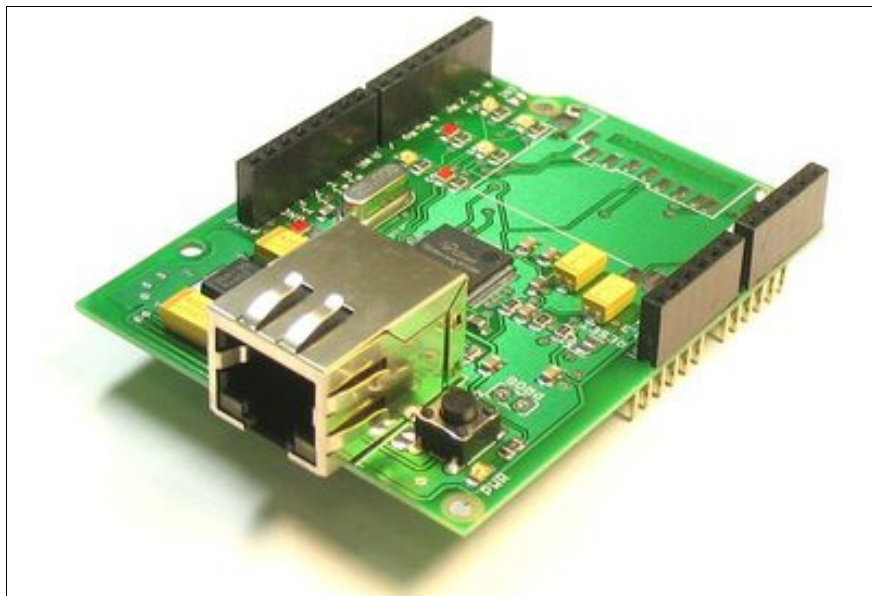


Рис. 3.2. Ethernet Shield

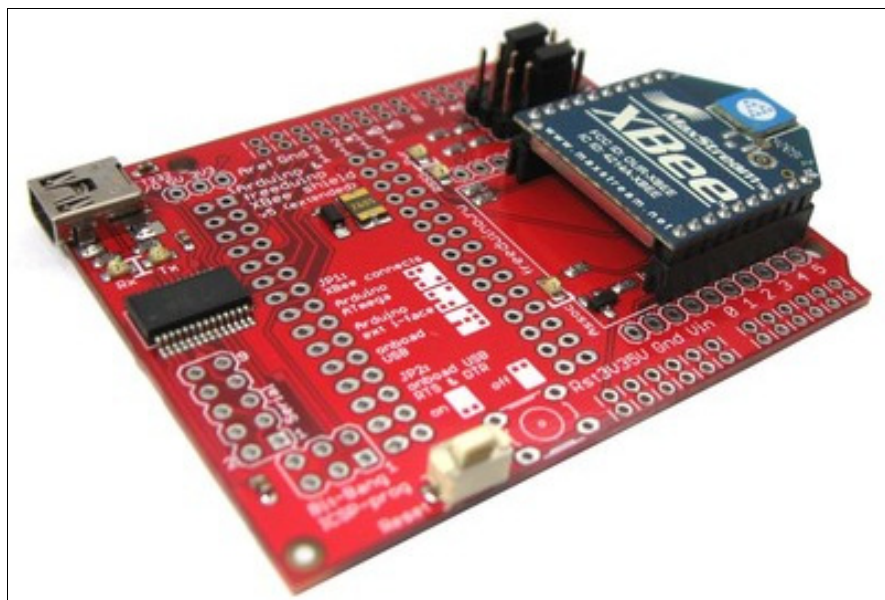


Рис. 3.3. XBee Shield

- **MicroSD Shield** (рис. 3.4) — обеспечивает запись данных на карты microSD;
- **MP3 Shield** (рис. 3.5) — плата для воспроизведения звука в форматах Ogg Vorbis/MP3/AAC/WMA/MIDI и записи в Ogg Vorbis;

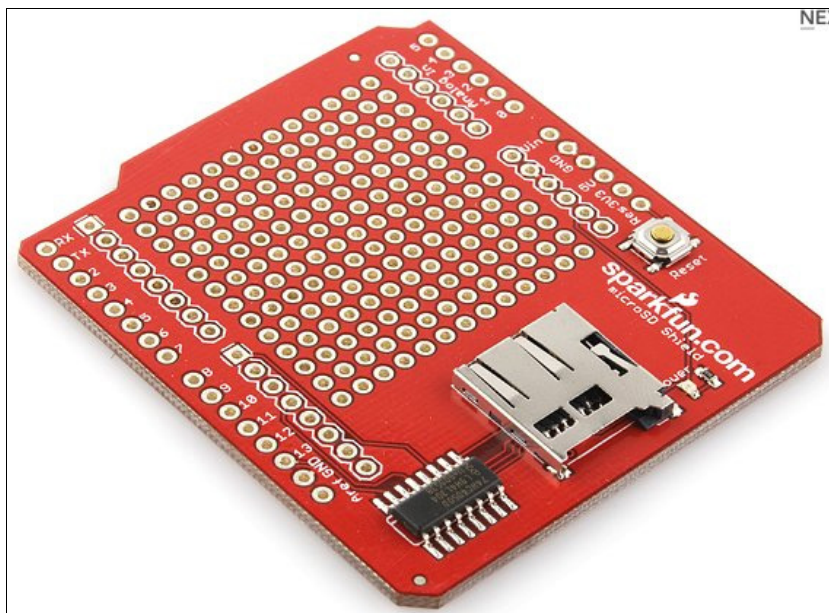


Рис. 3.4. MicroSD Shield

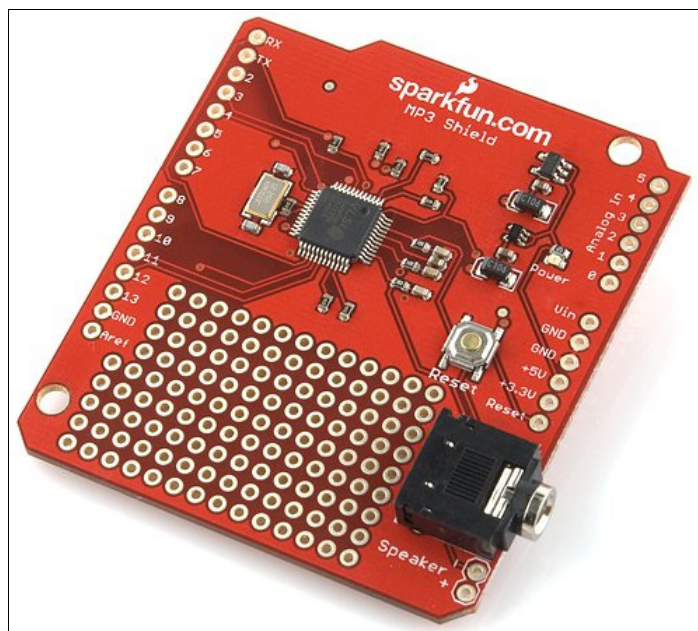


Рис. 3.5. MP3 Shield

- **Motor Shield** (рис. 3.6) — обеспечивает управление двигателями постоянного тока;
- **GSM/GPRS Shield** (рис. 3.7) — позволяет отправлять SMS-сообщения, делать звонки, обмениваться данными по GPRS;



Рис. 3.6. Motor Shield

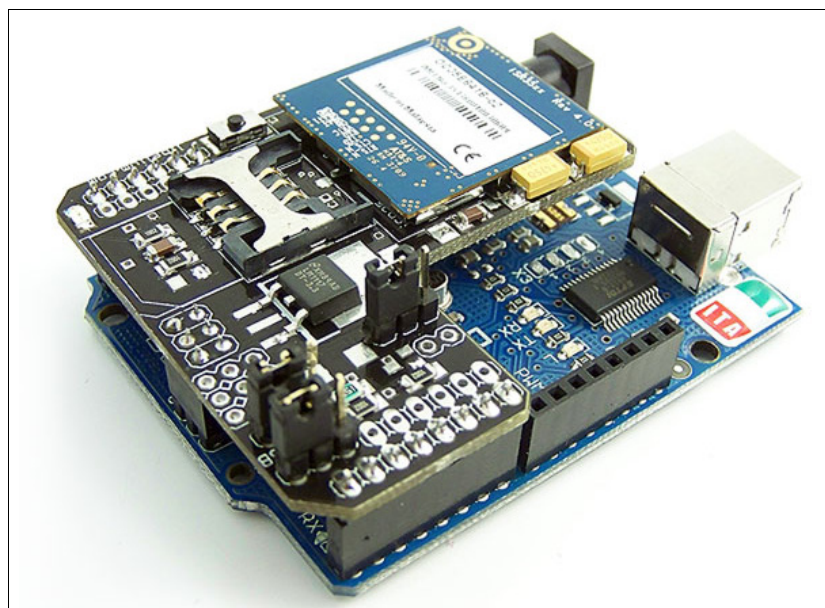


Рис. 3.7. GSM/GPRS Shield

□ **Cosmo WiFi Connect** (рис. 3.8) — плата предназначена для организации беспроводной сети стандарта IEEE 802.11b/g.

Существуют также шилды: **Video Overlay Shield** — для наложения текста на аналоговое видео, **EasyVR Arduino Shield** — многоцелевой модуль распознавания речи, **Music Shield** — профессиональный аудиокодек и др.

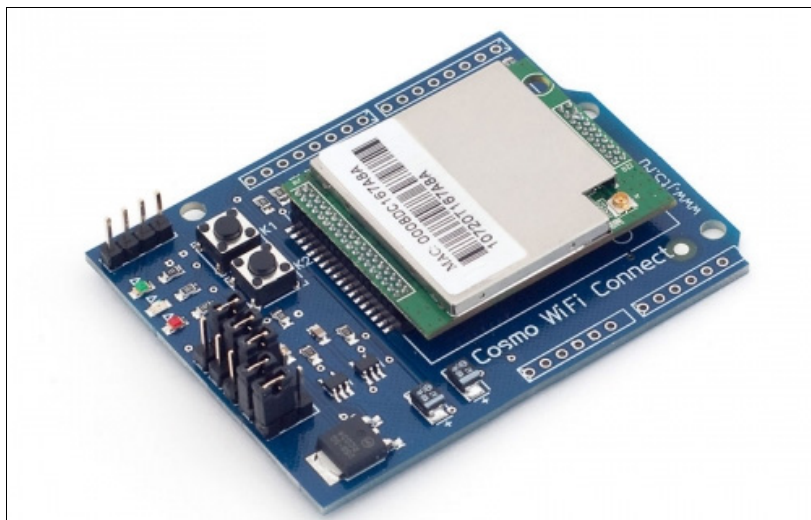


Рис. 3.8. Cosmo WiFi Connect

Количество плат расширения (шилдов) постоянно растет. Ознакомиться с их списком можно на официальном сайте проекта Arduino по адресу <http://www.arduino.cc/playground/Main/SimilarBoards#goShie>. Имеется также и еще один интересный ресурс: <http://shieldlist.org/>, где представлено описание 244 шилдов для Arduino.



---

# ЧАСТЬ II

---



## Среда разработки и язык программирования контроллеров Arduino

**Глава 4.** Среда программирования Arduino IDE

**Глава 5.** Программирование в Arduino





## ГЛАВА 4

# Среда программирования Arduino IDE

Разработка собственных приложений на базе плат, совместимых с архитектурой Arduino, осуществляется в официальной 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. Существуют версии среды для операционных систем Windows, Mac OS и Linux.

Последнюю версию среды Arduino 1.0.5 и бета-версию Arduino 1.5.2 (с поддержкой Arduino Due) можно скачать со страницы загрузки официального сайта <http://arduino.cc/en/Main/Software>.

### 4.1. Установка Arduino IDE в Windows

Отправляемся на страницу <http://arduino.cc/en/Main/Software> (рис. 4.1), выбираем версию для операционной системы Windows и скачиваем архивный файл. Он занимает чуть более 80 Мбайт и содержит все необходимое, в том числе и драйверы. По окончании загрузки распаковываем скачанный файл в удобное для себя место.

Теперь необходимо установить драйверы. Подключаем Arduino к компьютеру. На контроллере должен загореться индикатор питания — зеленый светодиод. Windows начинает попытку установки драйвера, которая заканчивается сообщением **Программное обеспечение драйвера не было установлено**.

Открываем Диспетчер устройств. В составе устройств находим значок Arduino Uno — устройство отмечено восклицательным знаком. Щелкаем правой кнопкой мыши на значке Arduino Uno и в открывшемся окне выбираем пункт **Обновить драйверы** и далее пункт **Выполнить поиск драйверов на этом компьютере**. Указываем путь к драйверу — ту папку на компьютере, куда распаковывали скачанный архив. Пусть это будет папка drivers каталога установки Arduino — например, C:\arduino-1.0\drivers. Игнорируем все предупреждения Windows и получаем в результате сообщение **Обновление программного обеспечения для данного устройст-**

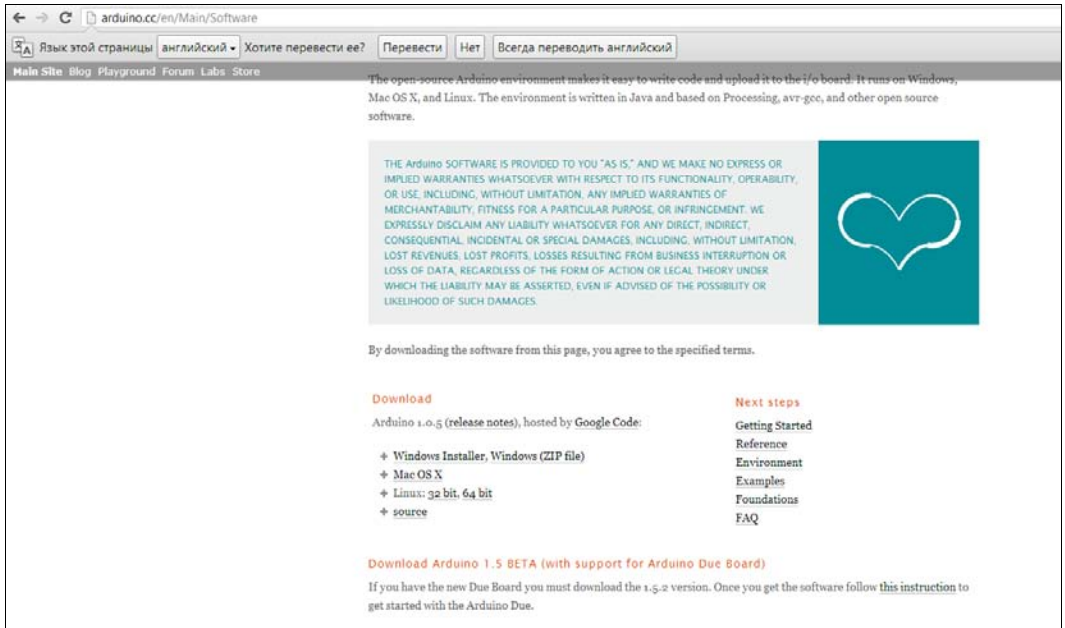


Рис. 4.1. Страница загрузки официального сайта Arduino

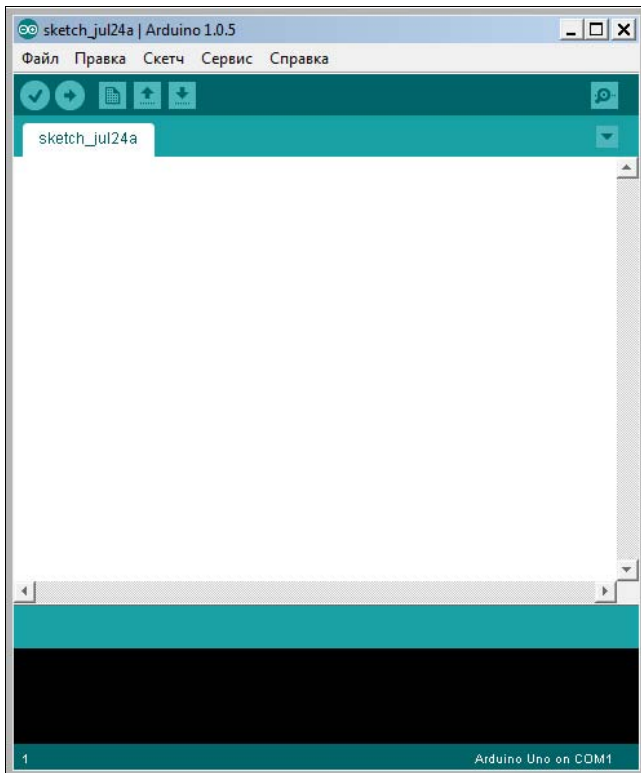


Рис. 4.2. Arduino IDE — среда разработки