

УДК 004.9Arduino, 004.732  
ББК 32.816, 32.973, 32.965  
И15

**Ибрагим Д.**

И15 Изучаем Arduino Uno R4 / пер. с англ. Ю. В. Ревича. – М.: ДМК Пресс, 2024. – 324 с.: ил.

**ISBN 978-5-93700-297-6**

Знакомая нам уже много лет плата Arduino Uno R3 на основе недорогого 8-битного процессора ATmega328P – возможно, самый популярный представитель семейства Arduino. Недавно была выпущена новая плата Arduino Uno R4 на основе 48-мегагерцевого 32-битного процессора Cortex-M4 с большим объемом SRAM и флеш-памяти. В конструкцию добавлены ЦАП и более точный АЦП, а также интерфейс CAN Bus. Доступны две версии новой платы: Uno R4 Minima и Uno R4 WiFi.

В книге рассказывается об использовании этих новых плат для разработки множества различных проектов с применением всего лишь нескольких деталей и внешних модулей. Для всех проектов предоставлены блок-схемы, принципиальные схемы, полные листинги программ на языке C и их краткие описания.

Вы познакомитесь с различными датчиками, разберетесь с устройством 7-сегментных индикаторов и ЖК-дисплеев, построите RFID-считыватель и часы реального времени (RTC), внедрите в свои конструкции сервоприводы и, наконец, узнаете, как связать устройства по Wi-Fi, Bluetooth, протоколу TCP/IP, через последовательный порт UART и CAN-шину.

Издание предназначено широкому кругу читателей, знакомых с Arduino и имеющих опыт работы с микроконтроллерами.

УДК 004.9Arduino, 004.732  
ББК 32.816, 32.973, 32.965

This is an Elektor Publication. Elektor is the media brand of Elektor International Media B.V. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-3-89576-578-0 (англ.)  
ISBN 978-5-93700-297-6 (рус.)

© 2023 Elektor International Media B.V.  
© Перевод, оформление, издание,  
ДМК Пресс, 2024

# Содержание

От издательства .....	10
Предисловие .....	11
<b>Глава 1 • Arduino Uno R4</b> .....	13
1.1. Обзор .....	13
1.2. Arduino Uno R4 в сравнении с Uno R3 .....	14
1.3. Аппаратное обеспечение Arduino Uno R4 Minima.....	17
<b>Глава 2 • Программирование Arduino Uno R4</b> .....	22
2.1. Обзор .....	22
2.2. Установка Arduino IDE 2.1.0 .....	23
2.3. Программы без использования аппаратной части.....	26
Пример 2.1. Сумма целых чисел .....	26
Пример 2.2. Таблица квадратов .....	28
Пример 2.3. Объем цилиндра.....	29
Пример 2.4. Перевод градусов Цельсия в градусы Фаренгейта .....	31
Пример 2.5. Таблица умножения.....	32
Пример 2.6. Таблица синуса .....	33
Пример 2.7. Таблица синуса, косинуса и тангенса .....	35
Пример 2.8. Целочисленный калькулятор .....	37
Пример 2.9. Игральные кубики .....	38
Пример 2.10. Калькулятор для действительных чисел .....	40
Пример 2.11. Вывод в двоичном, восьмеричном, шестнадцатеричном форматах.....	42
Пример 2.12. Строковые функции .....	43
Пример 2.13. Инициализация массива.....	44
Пример 2.14. Символьные функции .....	45
Пример 2.15. Решение квадратного уравнения .....	47
Пример 2.16. Счастливый день недели.....	50
Пример 2.17. Факториал числа.....	51
Пример 2.18. Сложение двух квадратных матриц.....	53
<b>Глава 3 • Аппаратные проекты со светодиодами</b> .....	55
3.1. Обзор .....	55
3.2. Проекты со светодиодами .....	55
Проект 3.1. Мигание встроенным светодиодом .....	55
Проект 3.2. Мигание внешним светодиодом.....	56
Проект 3.3. Мигающий светодиод SOS .....	60
Проект 3.4. Поочередное мигание светодиодов .....	61
Проект 3.5. Гонки светодиодов .....	64
Проект 3.6. Гонки светодиодов-2.....	67

Проект 3.7. Светодиодный двоичный счетчик .....	68
Проект 3.8. Случайное мигание светодиодов: елочная гирлянда .....	70
Проект 3.9. Светодиод, управляемый кнопкой .....	72
Проект 3.10. Управление частотой мигания светодиода через внешние прерывания .....	76
Проект 3.11. Измерение реакции .....	80
Проект 3.12. Цветная волшебная палочка.....	81
Проект 3.13. Заданные цвета RGB .....	84
Проект 3.14. Светофор.....	85
Проект 3.15. Светофор с пешеходным переходом .....	90
Проект 3.16. Сдвиговый регистр 74НС595: двоичный счетчик .....	96
Проект 3.17. Сдвиговый регистр 74НС595: случайное мигание 8 светодиодов .....	100
Проект 3.18. Сдвиговый регистр 74НС595: гонки светодиодов.....	101
Проект 3.19. Сдвиговый регистр 74НС595: включение заданного светодиода .....	102
Проект 3.20. Сдвиговый регистр 74НС595: демонстрация заданного числа на светодиодах .....	103
<b>Глава 4 • 7-сегментные светодиодные индикаторы .....</b>	<b>105</b>
4.1. Обзор.....	105
4.2. Структура 7-сегментного светодиодного индикатора .....	105
Проект 4.1. 7-сегментный одноразрядный индикатор: десятичный счетчик .....	107
Проект 4.2. Четырехразрядный 7-сегментный мультиплексированный светодиодный индикатор .....	111
Проект 4.3. Четырехразрядный 7-сегментный индикатор: счетчик с прерываниями таймера .....	117
Проект 4.4. Четырехразрядный 7-сегментный индикатор: счетчик с очисткой ведущих нулей.....	121
Проект 4.5. Четырехразрядный 7-сегментный индикатор: измерение реакции с помощью таймера .....	124
Проект 4.6. Мигание встроенного светодиода по прерыванию таймера .....	129
<b>Глава 5 • Жидкокристаллические дисплеи .....</b>	<b>131</b>
5.1. Обзор.....	131
5.2. Шина I2C .....	132
5.3. Порты I2C платы Uno R4 .....	132
5.4. ЖК-дисплей с интерфейсом I2C .....	133
Проект 5.1. Отображение текста на ЖК-дисплее .....	137
Проект 5.2. Прокрутка текста на ЖК-дисплее .....	138
Проект 5.3. Отображение пользовательских символов на ЖК-дисплее.....	140
Проект 5.4. Счетчик товаров на конвейерной ленте с ЖК-дисплеем .....	142
Проект 5.5. Точные часы на ЖК-дисплее с использованием прерываний таймера.....	145
Проект 5.6. Игральные кубики на ЖК-дисплее .....	150

<b>Глава 6 • Датчики</b> .....	153
6.1. Обзор .....	153
6.2. Проекты с различными датчиками .....	153
Проект 6.1. Аналоговый датчик температуры .....	153
Проект 6.2. Вольтметр .....	156
Проект 6.3. Релейный регулятор температуры .....	158
Проект 6.4. Фоторезистор в качестве датчика освещенности .....	161
Проект 6.5. Датчик наклона .....	164
Проект 6.6. Измерение уровня воды .....	166
Проект 6.7. Регулятор уровня воды .....	169
Проект 6.8. Датчик переполнения с зуммером .....	171
Проект 6.9. Датчик обнаружения звука – управление реле хлопком ладоней .....	173
Проект 6.10. Датчик пламени с релейным выходом .....	175
Проект 6.11. Измерение температуры и влажности .....	177
Проект 6.12. Генерация музыкальных тонов .....	181
<b>Глава 7 • RFID-считыватель</b> .....	184
7.1. Обзор .....	184
7.2. Применение RFID-считывателя .....	185
Проект 7.1. Определение идентификатора RFID-метки .....	185
Проект 7.2. Дверной замок с RFID-доступом через реле .....	187
<b>Глава 8 • Клавиатура 4×4</b> .....	191
8.1. Обзор .....	191
8.2. Кнопочная клавиатура 4×4 .....	191
Проект 8.1. Отображение нажатого кода клавиши на последовательном мониторе .....	192
Проект 8.2. Целочисленный калькулятор с ЖК-дисплеем .....	195
Проект 8.3. Дверной замок с клавиатурой и реле .....	201
<b>Глава 9 • Модуль часов реального времени (RTC)</b> .....	204
9.1. Обзор .....	204
9.2. Модуль RTC .....	204
Проект 9.1. RTC и монитор последовательного порта .....	205
Проект 9.2. RTC с ЖК-дисплеем .....	208
Проект 9.3. Отображение температуры и влажности с отметкой времени .....	210
9.3. Использование встроенных RTC контроллера Renesas RA4M1 .....	213
Проект 9.4. Установка и отображение текущего времени встроенных RTC .....	214
Проект 9.5. Периодическое прерывание каждые 2 секунды .....	215
<b>Глава 10 • Джойстик</b> .....	217
10.1. Обзор .....	217
10.2. Джойстик .....	217
Проект 10.1. Чтение аналоговых значений с джойстика .....	218

<b>Глава 11 • Светодиодная матрица 8×8</b> .....	222
11.1. Обзор.....	222
11.2. Светодиодная матрица 8×8.....	222
Проект 11.1. Отображение фигур .....	223
<b>Глава 12 • Двигатели: сервоприводы и шаговые двигатели</b> .....	228
12.1. Обзор.....	228
12.2. Сервопривод .....	228
Проект 12.1. Тестовое вращение сервопривода .....	229
Проект 12.2. Серворазвертка.....	231
Проект 12.3. Сервопривод, управляемый джойстиком.....	232
12.3. Шаговый двигатель.....	234
Проект 12.4. Поворот двигателя по часовой стрелке и обратно .....	235
<b>Глава 13 • Цифроаналоговый преобразователь</b> .....	238
13.1. Обзор.....	238
Проект 13.1. Генерация прямоугольного сигнала амплитудой 2 В.....	238
13.2. Генерация синусоидального сигнала: библиотека AnalogWave.....	240
Проект 13.2. Генерация синусоидального сигнала .....	240
Проект 13.3. Генератор синуса с переменной частотой .....	242
Проект 13.4. Задание частоты синусоидального сигнала с помощью потенциометра.....	242
Проект 13.5. Генерация прямоугольного сигнала частотой 1 кГц и амплитудой 1 В .....	244
<b>Глава 14 • Использование EEPROM, Human Interface Device и ШИМ</b> .....	245
14.1. Обзор.....	245
14.2. Память EEPROM.....	245
14.3. Интерфейс взаимодействия с человеком (HID) .....	246
Проект 14.1. Запуск программ Windows с помощью кнопок .....	247
14.4. Широтно-импульсная модуляция (ШИМ) .....	250
Проект 14.2. Регулирование светодиодов с помощью ШИМ.....	253
<b>Глава 15 • Arduino Uno R4 WiFi</b> .....	255
15.1. Обзор.....	255
15.2. Светодиодная матрица .....	258
Проект 15.1. Использование светодиодной матрицы для создания изображения креста, метод 1 .....	258
Проект 15.2. Создание изображений установкой байтов .....	260
Проект 15.3. Использование светодиодной матрицы для создания изображения креста, метод 2.....	263
Проект 15.4. Анимация – отображение слова.....	265
15.3. Использование Wi-Fi .....	267
15.3.1. UDP и TCP.....	267
15.3.2. Связь по протоколу UDP .....	268
15.3.3. Связь по протоколу TCP.....	269
Проект 15.5. Управление встроенным светодиодом Arduino Uno R4 WiFi со смартфона с помощью UDP .....	270
15.4. Bluetooth.....	274

15.4.1. Bluetooth BLE .....	275
15.4.2. Модель программного обеспечения Bluetooth BLE .....	276
<b>Глава 16 • Последовательная связь .....</b>	<b>278</b>
16.1. Обзор .....	278
Проект 16.1. Получение температуры окружающей среды от Arduino Uno R3.....	280
<b>Глава 17 • Использование симулятора Arduino Uno .....</b>	<b>283</b>
17.1. Почему симуляция?.....	283
17.2. Симулятор Wokwi .....	285
Проект 17.1. Простое моделирование – мигающий светодиод.....	285
Проект 17.2. Отображение текста на ЖК-дисплее .....	287
Проект 17.3. ЖК-счетчик секунд.....	288
<b>Глава 18 • Шина CAN .....</b>	<b>290</b>
18.1. Обзор.....	290
18.2. Шина CAN .....	290
18.2.1. Оконечная нагрузка CAN-шины.....	291
18.2.2. Скорость передачи данных по CAN-шине .....	292
18.2.3. Длина кабельного шлейфа .....	293
18.2.4. Узел CAN-шины.....	293
18.2.5. Уровни сигнала шины CAN .....	293
18.2.6. Напряжение CAN_H .....	294
18.2.7. Напряжение CAN_L .....	295
18.2.8. Арбитраж шины .....	295
18.2.9. Шинный приемопередатчик.....	296
18.2.10. CAN-разъемы.....	296
18.3. Интерфейс CAN-шины Arduino Uno R4 .....	298
18.3.1. Приемопередатчик CAN-шины .....	298
18.4. Примеры использования CAN.....	300
Проект 18.1. Связь между Arduino Uno R4 WiFi и Arduino Uno R4 Minima по CAN-шине .....	300
Проект 18.2. Отправка показаний температуры по CAN-шине.....	304
<b>Глава 19 • Инфракрасный приемник и пульт дистанционного управления.....</b>	<b>309</b>
19.1. Обзор.....	309
19.2. Инфракрасный приемник.....	309
19.3. Инфракрасный пульт дистанционного управления .....	310
19.4. Работа ИК-системы дистанционного управления .....	310
Проект 19.1. Расшифровка кодов ИК-пульта управления .....	313
Проект 19.2. Дистанционное включение/выключение реле .....	315
Проект 19.3. Дистанционное управление шаговым двигателем.....	318
<b>Предметный указатель .....</b>	<b>322</b>

# Предисловие

Arduino – это система с открытым исходным кодом для создания проектов на основе микроконтроллеров, включающая системное оборудование, интегрированную среду разработки (IDE) и большое количество библиотек. Arduino поддерживается большим сообществом программистов, инженеров-электронщиков, энтузиастов и ученых. Существует несколько различных конструкций базовых плат Arduino. Хотя они предназначены для различных типов приложений, все они могут быть запрограммированы с использованием одной и той же IDE, и, как правило, программы можно переносить (портировать) между разными платами. Вероятно, это одна из причин популярности семейства Arduino. Arduino поддерживает большое количество программных библиотек для многих интерфейсных устройств, которые можно легко включить в ваши проекты. Использование этих библиотек делает программирование относительно простым и ускоряет время разработки, а также упрощает тестирование программ, поскольку большинство библиотек уже полностью протестированы и гарантированно работают.

Плата Arduino Uno R3 на сегодняшний день, вероятно, считается самым популярным членом семейства Arduino и используется уже много лет. Основанная на недорогом 8-битном процессоре ATmega328P, Uno R3 применяется студентами и любителями во многих низкоскоростных проектах начального и среднего уровней, требующих малого и среднего объема памяти. Одним из привлекательных моментов применения Uno R3 стала мощная среда разработки и простота ее использования для разработки проектов в относительно короткие сроки.

Недавно была анонсирована новая Arduino Uno R4<sup>1</sup>. Эта плата совместима с более ранней версией Uno R3, но ее характеристики значительно улучшены. Arduino Uno R4 основана на 32-битном процессоре Cortex-M4 с тактовой частотой 48 МГц, содержит большой объем оперативной памяти SRAM и программной флеш-памяти. Кроме того, в конструкцию добавлены более точный аналого-цифровой преобразователь (АЦП) и ранее отсутствовавший цифроаналоговый преобразователь (ЦАП). Плата также поддерживает интерфейс CAN-шины. Доступны две версии платы: Uno R4 Minima и Uno R4 WiFi.

Эта книга посвящена использованию плат Uno R4 для множества различных интересных проектов. Проекты, приведенные в книге, были полностью протестированы с использованием ряда внешних компонентов и модулей. Для всех проектов в книге предоставлены блок-схемы, принципиальные схемы, полные листинги программ и их краткие описания. Вам будет несложно собрать необходимое аппаратное обеспечение для проектов, а затем следовать приведенным описаниям программ. Также должно быть относительно просто модифицировать аппаратное и программное обеспечение для ваших собственных приложений.

---

<sup>1</sup> Arduino Uno R4 в обеих версиях (Minima и WiFi) появилась в продаже в конце июня 2023 года. – *Здесь и далее прим. перев.*

Я надеюсь, что вам понравится книга и с ее помощью вы научитесь использовать модели Arduino Uno R4 Minima или R4 WiFi в своих инновационных проектах.

Доган Ибрагим,  
Лондон, 2023 г.

**Примечание издателя:** все программы, обсуждаемые в этом руководстве, содержатся в архивном файле, который вы можете скачать бесплатно с сайта Elektor. Зайдите на сайте [www.elektor.com](http://www.elektor.com) в раздел «Books» и введите название книги в поле поиска.

## Текстовые соглашения

В этой книге приняты следующие текстовые соглашения.

Код в тексте: означает кодовые слова и имена переменных в тексте.

Блок кода задается следующим образом:

```
#include <LCD_I2C.h>
LCD_I2C lcd(0x27, 16, 2);
void setup()
{
  lcd.begin();
}
```

*Курсивное начертание* означает имена файлов, имена папок, расширения файлов, пути файловой системы, названия программ и библиотек. Примеры: «...на момент написания этой книги имя файла последней версии было: *Arduino-ide\_2.1.0-Windows\_64bit.exe*». Курсивным начертанием также могут выделяться отдельные слова или целые фразы, на которые следует обратить внимание.

**Жирный шрифт:** обозначает названия экранных кнопок и пунктов меню. Примеры: «...с помощью пункта меню **Edit** (*Редактировать*)...»; «Нажмите кнопку **Quit** (*Выход*)».

Названия клавиш на клавиатуре и их сочетаний также выделяются жирным шрифтом и заключаются в угловые скобки: **<Ctrl+Q>**.



# Глава 1 • Arduino Uno R4

## 1.1. Обзор

Проект Arduino начался как создание инструмента для студентов Института интерактивного дизайна города Ивреа в Италии еще в 2005 году. Целью этого проекта было предоставить начинающим студентам и любителям недорогое и простое в использовании аппаратное и программное обеспечение для создания простых проектов с использованием датчиков и исполнительных механизмов. Название «Arduino» происходит от названия бара «Arduin of Ivrea», где основатели проекта собирались для обсуждения. Название этого бара произошло от имени маркграфа Ардуина из Ивреа, который был королем Италии с 1002 по 1014 год.

Первоначальная команда проекта Arduino состояла из Массимо Банзи (Massimo Banzi), Дэвида Куартъелеса (David Cuartielles), Тома Иго (Tom Igoe), Джанлуки Мартино (Gianluca Martino) и Дэвида Меллиса (David Mellis). В 2003 году Эрнандо Барраган (Hernando Barragan) из института Ивреа создал платформу разработки Wiring в качестве магистерской диссертации. Wiring представляла собой электронную платформу с открытым исходным кодом, состоящую из языка программирования, интегрированной среды разработки (IDE) и одноплатного микроконтроллера. Этот проект был разработан под руководством Массимо Банзи и Кейси Риса (Casey Reas). Платформа Wiring включала печатную плату с микроконтроллером ATmega128, IDE и некоторые библиотечные функции. Позже, в 2005 году, Массимо Банзи, Дэвид Меллис и Дэвид Куартиелес расширили платформу Wiring, добавив поддержку микроконтроллера ATmega8, который был дешевле. Этот новый проект получил название Arduino. Проект оказался настолько успешным, что после разработки менее дорогих версий к середине 2011 года было коммерчески произведено более 300 000 только оригинальных плат Arduino, а к 2013 году это число увеличилось до более чем 800 000.

Arduino – это аппаратное обеспечение с открытым исходным кодом, проекты которого распространяются по лицензии Creative Commons и находятся в свободном доступе. Единственный момент заключается в том, что разработчикам клонов предлагается зарезервировать имя Arduino для официального продукта и не использовать его для аналогов без разрешения.

Одной из приятных особенностей серии Arduino является то, что на встроенном процессоре используется предварительно запрограммированный загрузчик. Пользователи могут разрабатывать свои программы с помощью IDE, а затем загружать свои программы в плату с контроллером Arduino посредством этой программы-загрузчика. Контакты ввода-вывода доступны в гнездовых разъемах, расположенных по обе стороны платы. Это значительно упрощает разработку аппаратного обеспечения, поскольку для подключения к плате можно использовать гибкие перемычки.

Оригинальная плата Arduino производилась итальянской компанией Smart Projects. Различными компаниями протяжении многих лет были разработаны другие версии Arduino. Некоторые из версий следующие:

- Arduino Diecimila;
- Arduino Uno R2;
- Arduino Leonardo;
- Arduino RS232;
- Arduino Pro;
- Arduino Mega;
- Arduino LilyPad;
- Arduino Robot;
- Arduino Esplora;
- Arduino Yun;
- Arduino Fio;
- Arduino Ethernet;
- Arduino Due;
- Arduino Nano;
- Arduino Uno SMD R3;
- Arduino Uno R3;
- Arduino MKR Zero;
- Arduino Zero
- ... и многие другие.

Семейство Arduino оказалось настолько популярным, что в 2022 году выручка этого сектора рынка составила более 237 млн долл. США, причем большую часть составили онлайн-продажи через интернет<sup>1</sup>.

Недавно (в конце июня 2023 года) были анонсированы две новые версии Arduino: Arduino Uno R4 Minima и Arduino Uno R4 WiFi. В этой книге вы будете разрабатывать проекты, используя как Arduino Uno R4 Minima, так и Arduino Uno R4 WiFi. Новые две платы похожи на популярную плату Arduino Uno R3, но они были во многом переработаны и расширены.

## 1.2. Arduino Uno R4 в сравнении с Uno R3

Сравнение Uno R3 и Uno R4 приведено в табл. 1.1. Обратите внимание, что почти все проекты и библиотеки, используемые с Uno R3, можно применять с Uno R4 без каких-либо изменений. Однако рекомендуется обновить любые библиотеки, которые могли быть переработаны специально для Uno R4.

---

<sup>1</sup> Следует отметить, что для большинства популярных версий Arduino существуют более дешевые аналоги, выпускающиеся в основном китайскими компаниями, есть и разработанные в России. Как правило, они носят то же или похожее название, но могут отличаться в схемотехнических деталях (и необязательно в худшую сторону). Такие аналоги получили большое распространение в России, но следует предостеречь от приобретения китайских аналогов от неизвестных или малоизвестных поставщиков и производителей (как представленных в мелких российских интернет-магазинах, так и на AliExpress), среди продукции которых много откровенного брака, непригодного к применению. То же самое относится и к внешним модулям Arduino.

Таблица 1.1. Сравнение плат Arduino Uno R3 и Arduino Uno R4

Компонент	Arduino Uno R3	Arduino Uno R4
Процессор	ATmega328P	Renesas RA4M1
Длина слова	8	32
Тактовая частота	16 МГц	48 МГц
SRAM	2 Кбайта	32 Кбайта
Flash-память	32 Кбайта	256 Кбайт
EEPROM	1 Кбайт	8 Кбайт
Рабочее напряжение	5 В	5 В
Таймеры	3	10
Емкостной сенсор	Нет	27 каналов
Датчик температуры	Нет	1
USB-разъем	Type B	USB-C
Аналогово-цифровой преобразователь (АЦП)	10 бит	14 бит
Цифроаналоговый преобразователь (ЦАП)	Нет	12 бит
Интерфейс SPI	1	1
Интерфейс I2C	1	2
Интерфейс Qwiic I2C	Нет	1 (только версия WiFi)
Последовательный порт UART	1	1
Wi-Fi	Нет	Только версия WiFi
Часы реального времени RTC	Нет	1 (только версия WiFi)
Интерфейс HID	Нет	Да
Отладка SWD debug	Нет	1
Интерфейс Bluetooth BLE	Нет	Только версия WiFi
Поддержка шины CAN	Нет	1
Операционный усилитель (ОУ)	Нет	1
Светодиодная матрица 128×8	Нет	Только версия WiFi
Напряжение питания	7–12 В	6–24 В
Аналоговые входы	6	6
Широтно-импульсная модуляция (PWM)	6 выводов	6 выводов
Тип USB	USB-B	USB-C

Плата Arduino Uno R4 оснащена центральным процессором Renesas RA4M1, представляющим собой 32-битный процессор Arm Cortex-M4, работающий на частоте 48 МГц. Uno R3 имела процессор ATmega328P с тактовой частотой всего 16 МГц, что дает трехкратное увеличение тактовой частоты по сравнению с Uno R3. Кроме того, Uno R4 имеет 32 Кбайта SRAM памяти по сравнению с 2 Кбайтами на Uno R3. Объем флеш-памяти R4 составляет 256 Кбайт по сравнению с 32 Кбайтами у Uno R3. В результате с помощью Uno R4 можно разрабатывать более сложные проекты, требующие большего объема памяти.

Разъем USB-B на Uno R3 был заменен на Uno R4 стандартным в настоящее время разъемом USB-C, а максимально допустимое напряжение внешнего источника питания было увеличено до 24 В с улучшением теплоотвода. Рабочее напряжение процессора по-прежнему составляет 5 В. Arduino Uno R4 предоставляет интерфейс CAN-шины, позволяющий подключать и про-

граммировать CAN-устройства. Количество интерфейсов шины SPI и I2C увеличено с 1 до 2.

Uno R4 поддерживает интерфейс взаимодействия с человеком HID (Human Interface Device), который позволяет пользователям имитировать мышь или клавиатуру при подключении к компьютеру через USB-кабель. Пользователи могут отправлять движения мыши или нажатия клавиш на компьютер.

На Uno R4 разрешение аналого-цифрового преобразователя (АЦП) увеличено с 10 до 14 бит. Кроме того, Uno R4 включает в себя настоящий 12-битный ЦАП. Для упрощения использования ЦАП была добавлена библиотека *AnalogWave*. Генерировать синусоидальный, пилообразный или прямоугольный сигнал можно простым вызовом библиотечной функции. Конечно, с помощью этой библиотеки вы можете сделать гораздо больше.

Печатная плата Uno R4 аппаратно совместима с Uno R3. Разводка выводов, напряжение и форм-фактор не изменились, поэтому Uno R4 может легко заменить конструкции, в которых используется Uno R3. Программное обеспечение IDE также совместимо между Uno R3 и Uno R4, при этом была предпринята попытка максимизировать обратную совместимость библиотек Arduino, чтобы пользователи могли применять существующие библиотеки без каких-либо изменений. Некоторые библиотеки, которые сильно зависят от AVR-архитектуры, возможно, потребуются заменить в комплектации IDE Arduino. Arduino должна предоставить список таких библиотек.

По сравнению с другими платами на базе Cortex, такими как Raspberry Pi Pico, в которой используется Cortex M0+, производительность Cortex-M4 примерно в 6 раз выше (напоминаем, что тактовая частота Raspberry Pi Pico по умолчанию равна 125 МГц). В результате Arduino Uno R4 можно использовать в очень сложных проектах реального времени (например, цифровое управление, DSP-процессоры, искусственный интеллект и т. д.), требующих большого объема памяти и высокой производительности.

Как уже говорилось ранее, Arduino Uno R4 доступен в двух версиях: WiFi и Minima. Версия Wi-Fi оснащена модулем Wi-Fi Espressif S3<sup>1</sup>, что делает плату идеальной для сетей «интернета вещей» (IoT) и приложений Bluetooth BLE, а также для приложений, требующих подключения к интернету. Дополнительно на плате установлена светодиодная матрица 128×8. Версия Minima предлагает экономичный вариант без подключения к Wi-Fi или Bluetooth. *В этой книге все проекты компилируются и запускаются в обеих версиях, за исключением проектов с использованием WiFi и LED Matrix, которые компилируются и запускаются только в версии Uno R4 WiFi.*

Подводя итог, можно сказать, что Uno R4 – это гигантский шаг вперед для Arduino, поскольку это действительно замечательная плата, которая поднимет ваш опыт работы с микроконтроллерами на новый уровень. Пожалуй, одним из недостатков Uno R4 по сравнению с Uno R3 является его повышенная цена.

---

<sup>1</sup> Модуль Espressif S3 производится компанией Espressif, известной своим семейством ESP8266/ESP32.

## 1.3. Аппаратное обеспечение Arduino Uno R4 Minima

На рис. 1.1 показан внешний вид печатной платы Arduino Uno R4. Посадочный размер печатной платы и расположение разъемов такие же, как у Uno R3. В центре печатной платы расположен микроконтроллер RA4M1. На левой (по рисунку) стороне печатной платы можно увидеть кнопку Reset, разъем USB-C для подключения к компьютеру и круглый гнездовой разъем DC для подачи внешнего питания. На противоположном краю печатной платы вы можете увидеть 6-контактный разъем внутрисхемного программирования (In System Programming, ISP, он же интерфейс SPI) и 10-контактные контакты интерфейса SWD/JTAG. Встроенный светодиод подключен к выводу 13, как и в Uno R3. Кроме того, к контактам последовательного порта TX и RX подключены желтые светодиоды для индикации последовательной передачи данных, зеленый светодиод указывает на питание платы, а желтый светодиод указывает на состояние линии SCK интерфейса I2C.

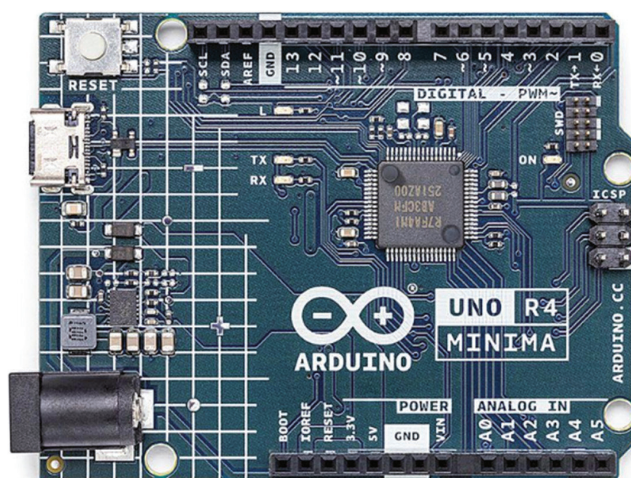


Рис. 1.1. Внешний вид печатной платы Arduino Uno R4 Minima

На двух краях печатной платы сверху и снизу расположены разъемы, как в Uno R3. Разъемы имеют следующие контакты, справа налево (см. рис. 1.2).

**Короткий разъем** (на рис. 1.1 снизу):

- 6 выводов A0–A5: 14-битные аналоговые входы;
- вывод A0 12-битный выход DAC;
- выводы ОУ + (A1), – (A2), выход (A3);
- выводы I2C: SDA (A4), SCL (A5);
- Vin (вход питания);
- GND (общий провод);
- +5 V (выход питания);
- +3.3 V (от вывода VCC\_USB процессора RA4M1);
- RESET (вывод сброса);
- IOREF (уровень цифрового логического сигнала. Подключен к + 5 V);
- BOOT (выбор режима загрузки).

**Примечание.** Аналоговые выводы А0–А5 также можно использовать в качестве цифровых.

**Длинный разъем** (на рис. 1.1 сверху):

- 14 цифровых выводов (D0–D13);
- два внешних прерывания (IRQ00: D2, IRQ01: D3);
- 6 выводов PWM (D3, D5, D6, D9, D10, D11);
- интерфейс UART (RX:D0, TX:D1);
- интерфейс SPI (те же выводы на разъеме ICSP: SCK: D13, MISO (CIPO): D12, MOSI (COPI): D11, CS: D10);
- интерфейс CAN (на Minima: RX: D5, TX: D4, на WiFi: RX: D13, TX:D10).  
Примечание: требуется внешний приемопередатчик;
- GND (общий провод);
- AREF (аналоговое опорное напряжение, по умолчанию подключен к +5V через резистор 5.1 кОм);
- I2C вывод SDA (подтяжка к питанию не установлена);
- I2C вывод SCL (подтяжка к питанию не установлена).

**ICSP-разъем:**

- см. табл. 1.2.

**Разъем SWD/JTAG:**

- см. табл. 1.3.

Таблица 1.2. Выводы разъема ICSP

Вывод	Функция	Тип	Описание
1	CIPO (MISO)	Internal	Controller In Peripheral Out (вход контроллера, выход периферии)
2	+5V	Internal	Питание 5 V
3	SCK	Internal	Тактовые импульсы
4	COPI (MOSI)	Internal	Controller Out Peripheral In (выход контроллера, вход периферии)
5	RESET	Internal	Сброс (Reset)
6	GND	Internal	Общий провод

Таблица 1.3. Выводы разъема SWD/JTAG

Вывод	Функция	Тип	Описание
1	+5V	Internal	Питание 5 V
2	SWDIO	Internal	Вход/выход данных
3	GND	Internal	Общий провод
4	SWCLK	Internal	Тактовые импульсы
5	GND	Internal	Общий провод
6	NC	Internal	Не подключен
7	RX	Internal	Последовательный прием
8	TX	Internal	Последовательная передача
9	GND	Internal	Общий вывод
10	NC	Internal	Не подключен

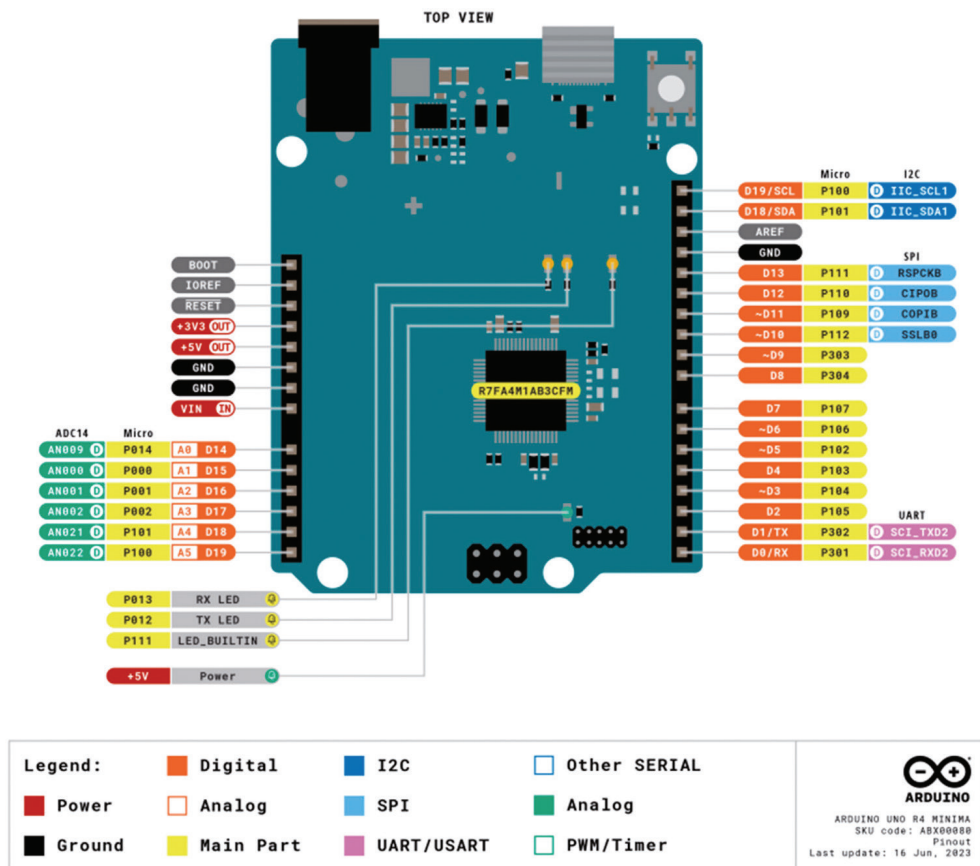


Рис. 1.2. Контакты разъемов Arduino Uno R4 Minima  
 (из «Справочного руководства», номер документа ABX0080)

Рекомендуемые условия эксплуатации приведены в табл. 1.4. На рис. 1.3 упрощенно показано подключение Arduino Uno R4 к источнику питания. Диоды Шоттки (находятся на плате) используются для защиты от перенапряжения и обратной полярности. Питание может подаваться через контакт VIN, круглый разъем питания DC или через разъем USB-C. Если питание подается через VIN (или круглый разъем), встроенный преобразователь понижает напряжение до 5 В. Питание через USB подает на микроконтроллер RA4M1 около 4,7 В (из-за прямого смещения на диоде Шоттки). Процессор RA4M1 может работать от +1,6 до +5,5 В и подключается к +5 В на плате Arduino. Цифровые выходы (GPIO) микроконтроллера RA4M1 могут выдерживать токи высокого уровня IOH и низкого уровня IOL = 8 мА. Помните, что IOL – это вытекающий (направленный к выводу) ток, когда вывод находится в состоянии логического 0, а IOH – это вытекающий (направленный от вывода) ток, когда вывод находится в состоянии логической 1. Необходимо соблюдать осторожность, чтобы не превышать рекомендуемый ток портов GPIO.

Таблица 1.4. Рекомендуемые условия эксплуатации

Обозначение	Описание	Мин.	Тип.	Макс.
$V_{IN}$	Напряжение питания через вывод VIN или круглый разъем DC, В	6	7,0	24
$V_{USB}$	Напряжение питания через USB-разъем, В	4,8	5,0	5,5
$t_{\text{раб}}$	Рабочая температура, °C	-40	25	85

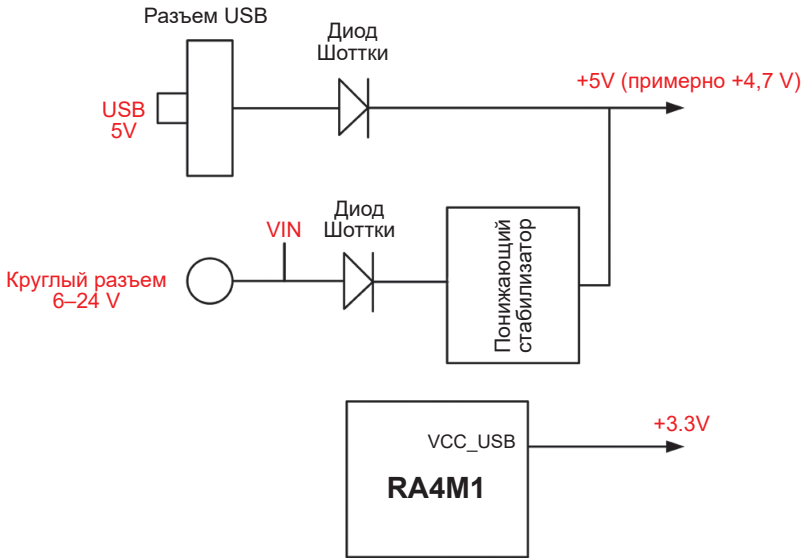


Рис. 1.3. Упрощенная схема подключения к источникам питания

На рис. 1.4 показано обозначение выводов платы Arduino Uno R4 Minima. Размещение компонентов дано на рис. 1.5 (рисунок взят из «Справочного руководства», документ AVX00080), а описания компонентов приведены в табл. 1.5.

Таблица 1.5. Компоненты на плате Arduino Uno R4 Minima

Название	Описание	Название	Описание
U1	R7FA4M1 AB3CFM#AA0 чип микроконтроллера	J4	DC разъем питания
U2	ISL854102FRZ-T понижающий преобразователь	DL1	LED TX (выход последовательного порта)
PB1	Кнопка RESET	DL2	LED RX (вход последовательного порта)
JANALOG	Аналоговые I/O выводы	DL3	LED питания
JDIGITAL	Цифровые I/O выводы	DL4	LED SCK (тактовые импульсы I2C)
J1	ISP разъем (SPI)	D2	Диод Шоттки PMEG6020AELRX
J2	Разъем SWD/JTAG	D3	Диод Шоттки PMEG6020AELRX
J3	Разъем CX90B-16P USB-C®	D4	PRTR5V0U2X,215 защита ESD



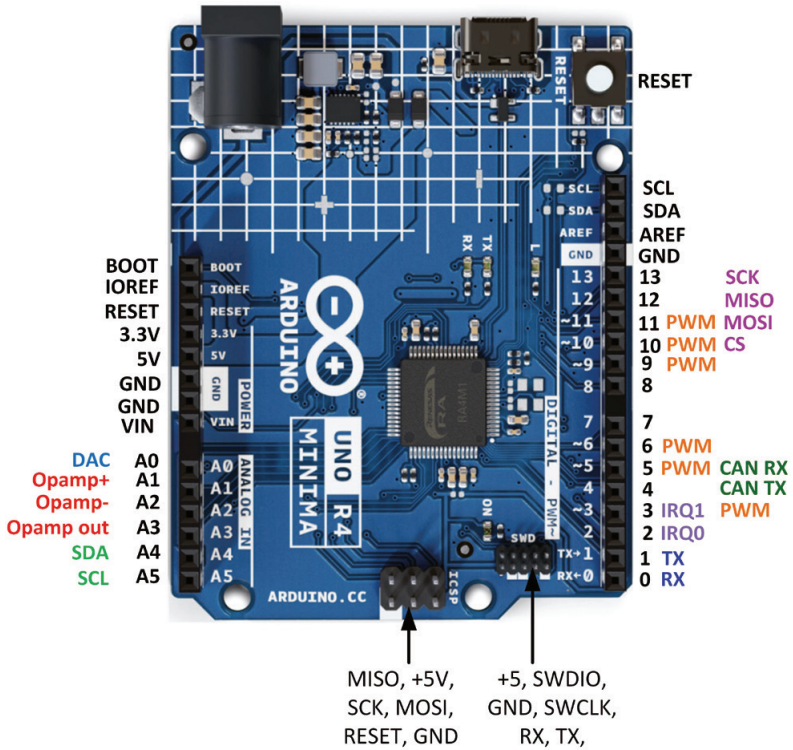


Рис. 1.4. Обозначение выводов Arduino Uno R4 Minima

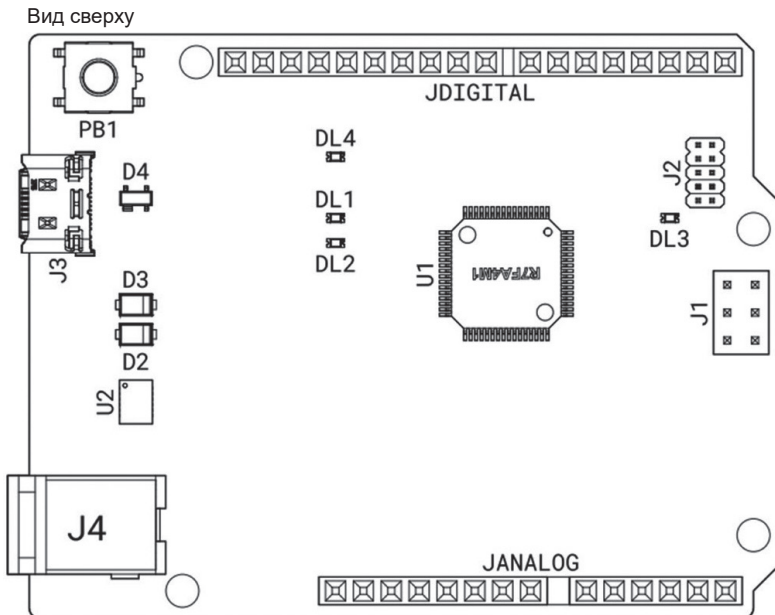


Рис. 1.5. Размещение компонентов Arduino Uno R4 Minima

# Глава 2 • Программирование Arduino Uno R4

## 2.1. Обзор

Код Arduino написан на языке C++, который в настоящее время является одним из самых популярных языков, используемых для программирования микроконтроллеров. Arduino можно программировать с помощью локального приложения Arduino IDE или веб-редактора Arduino Web Editor. Приложение Arduino IDE требует, чтобы программное обеспечение было установлено на вашем компьютере.

Интегрированная среда разработки (IDE) состоит из редактора, компилятора, отладчика и инструментов для загрузки скомпилированного кода в процессор на плате Arduino. Программа Arduino называется «скетчем», который компилируется в машинный код и загружается в целевой процессор. Веб-редактор Arduino позволяет программисту разрабатывать, компилировать и загружать программы с помощью браузера онлайн, причем этот способ имеет то преимущество, что онлайн-инструмент всегда актуален и включает в себя новейшие библиотеки и функции. В этой книге мы будем использовать локальное IDE-приложение. Заинтересованные читатели могут найти веб-редактор Arduino (см. табл. 2.1), создать учетную запись для входа и использовать его.

Современная Arduino IDE разрабатывалась более 10 лет, и существует несколько ее версий. Последняя стабильная версия – 2.1, выпущенная в марте 2022 года. Предыдущая версия 1.8.19 пользуется популярностью уже много лет и до сих пор используется многими программистами. Рекомендуется применять новую версию 2.1.0, поскольку она быстрее и проще в использовании, чем версия 1.8.19<sup>1</sup>. Автор использовал версию 2.1.0 во всех проектах этой книги. Читатели могут предпочесть работать с одной и той же версией в своих проектах.

В этой главе вы узнаете, как установить версию IDE 2.1.0, которая была последней версией на момент написания нашей книги. В данной главе с целью обзора принципов программирования с использованием Arduino IDE представлены простые программы, предназначенные только для программного обеспечения без подключения внешних модулей. В следующих главах вы будете использовать платы Arduino Uno R4 Minima и WiFi совместно с внешними компонентами.

Дополнительную информацию об Arduino IDE, веб-редакторе и сопутствующих инструментах можно получить по ссылкам, приведенным в табл. 2.1.

---

<sup>1</sup> Для установки Arduino IDE 2.x под Windows требуется версия Windows 10/64 бит или более новая. Все предыдущие версии, включая Arduino IDE 1.8.19, могут работать под Windows 7 или новее. Так как версии 2.x основательно переработаны, часто их обобщенно называют Arduino IDE 2, а предыдущие – Arduino IDE 1.

Таблица 2.1. Ссылки на Arduino IDE, веб-редактор и сопутствующие инструменты

Ресурс	Ссылка
Arduino IDE (настольное приложение)	<a href="https://www.arduino.cc/en/Main/Software">https://www.arduino.cc/en/Main/Software</a>
Arduino IDE (веб-приложение)	<a href="https://create.arduino.cc/editor">https://create.arduino.cc/editor</a>
Начиная работу с IDE	<a href="https://docs.arduino.cc/eloud/web-editor/tutorials/getting-started/getting-started-web-editor">https://docs.arduino.cc/eloud/web-editor/tutorials/getting-started/getting-started-web-editor</a>
Arduino Project Hub	<a href="https://create.arduino.cc/projecthub?by=part&amp;part id=11332&amp;sort=trending">https://create.arduino.cc/projecthub?by=part&amp;part id=11332&amp;sort=trending</a>
Справка по библиотекам Arduino Library	<a href="https://github.com/arduino-libraries/">https://github.com/arduino-libraries/</a>
Интернет-магазин Arduino Store	<a href="https://store.arduino.cc/">https://store.arduino.cc/</a>

## 2.2. Установка Arduino IDE 2.1.0

Последнюю версию Arduino IDE можно установить с официального сайта по ссылке <https://www.arduino.cc/en/software>.

Выберите свою операционную систему в разделе **Download Options** (*Опции загрузки*) справа (рис. 2.1). Далее нажмите **Just Download** (*Просто загрузите*), если вы не хотите поддержать взносом. Автор устанавливал приложение на ноутбук с Windows 10, на момент написания этой книги имя файла последней версии было: *Arduino-ide\_2.1.0-Windows\_64bit.exe*<sup>1</sup>.

### Downloads

Рис. 2.1. Выбор операционной системы для установки

После этого вам нужно установить пакет для платы Arduino Uno R4 Minima. По шагам:

- запустите Arduino IDE;
- нажмите **Boards Manager** (*Менеджер плат*<sup>2</sup>) в левом верхнем углу экрана (рис. 2.2);

<sup>1</sup> Версия 1.8.19 размещена ниже на той же странице. Для доступа к еще более старым версиям прокрутите страницу до раздела «Previous Releases» в самом низу. Языковую версию выбирать специально не требуется – язык меню выбирается в процессе установки и легко меняется при необходимости.

<sup>2</sup> Перевод пунктов меню Arduino IDE 2 здесь выполняется по смыслу и может не совпадать с официальным переводом в различных русскоязычных версиях.

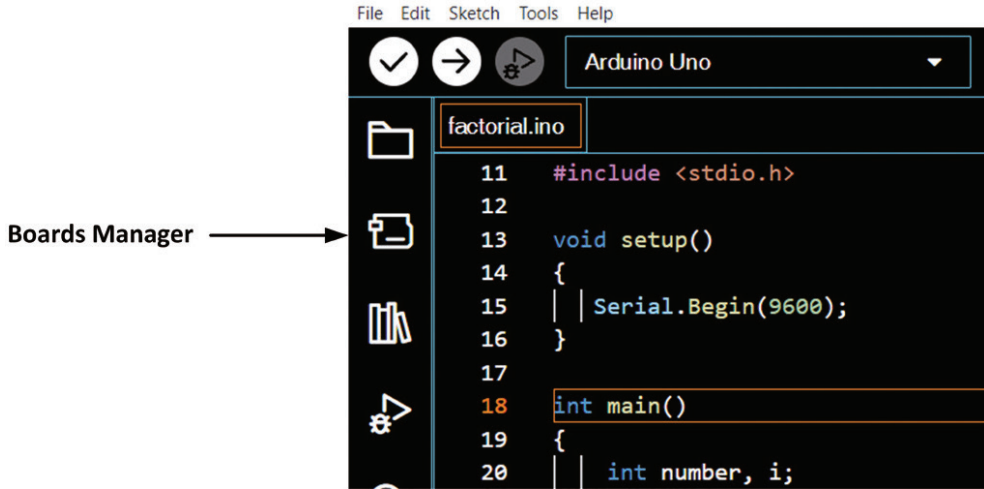


Рис. 2.2. Откройте **Boards Manager**

- найдите в списке **Arduino UNO R4** (рис. 2.3) и нажмите **INSTALL** (*Установить*) для установки. На момент написания этой книги номер версии пакета был 1.0.1;

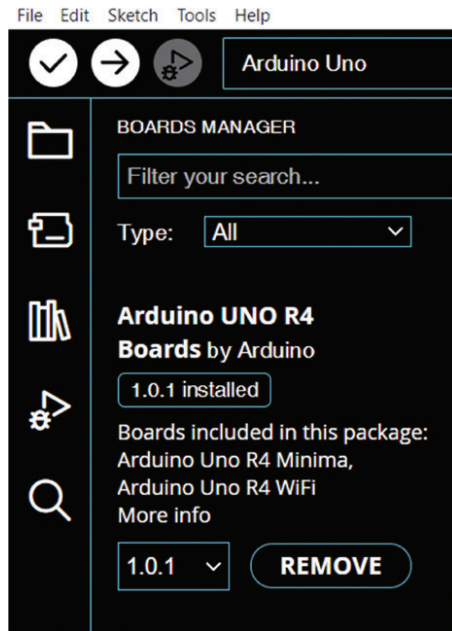


Рис. 2.3. Нажмите **INSTALL** для установки пакета Uno R4

- опять нажмите **Boards Manager** (*Менеджер плат*), чтобы закрыть окно слева;
- 24

- вы сможете выбрать плату Arduino Uno R4 с помощью переключателя плат в левом верхнем углу экрана (рис. 2.4). Подключите Arduino Uno R4 Minima (или Uno R4 WiFi) к компьютеру с помощью кабеля USB-C;
- нажмите **Tools** (*Инструменты*) → **Port** (*Порт*) и выберите последовательный порт, соответствующий вашему Uno R4.

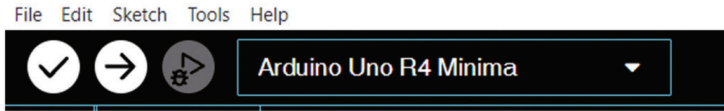


Рис. 2.4. Выберите Arduino Uno R4 Minima

Теперь вы готовы разрабатывать программы и загружать их в процессор на плате Arduino Uno R4 Minima/WiFi.

Прежде чем рассматривать примеры программ, остановимся на некоторых часто используемых разделах меню, предлагаемых Arduino IDE.

**File** (*Файл*): с помощью этого пункта меню вы можете открывать существующие программы, сохранять программы, открывать примеры программ и настраивать рабочую среду IDE через пункт **Preferences...** (*Настройки...*).

**Edit** (*Редактировать*): посредством этого пункта меню вы можете вырезать, вставлять, копировать фрагменты текста, переходить к указанной строке, изменять отступ и размер шрифта, а также находить текст в файле.

**Sketch** (*Скетч*): с помощью этого пункта меню вы можете скомпилировать свою программу, загрузить скомпилированную программу в целевой процессор, подключить библиотеки, добавить файлы. Есть и некоторые другие позиции, которые мы не будем здесь использовать.

**Tools** (*Инструменты*): посредством этого пункта меню вы можете управлять библиотеками, настраивать монитор последовательного порта (Serial Monitor) и построитель графиков (plotter), выбирать и настраивать плату, которую вы будете использовать, а также записывать в нее новый загрузчик.

**Help** (*Справка*): этот пункт меню отображает различные элементы справки по выбранным темам. Кроме того, он отображает номер версии, используемой в настоящий момент IDE (рис. 2.5). В данном случае это «Version: 2.1.0».

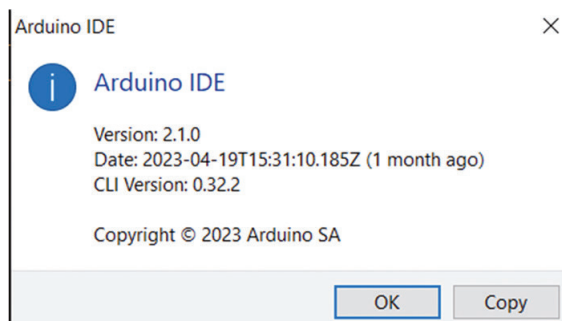


Рис. 2.5. Отображение номера версии IDE

## 2.3. Программы без использования аппаратной части

В этом разделе представлены простые программные примеры для обзора основных принципов программирования на языке C с использованием Arduino IDE. Человеческий интерфейс (к которому относятся, например, дисплей и клавиатура) для этих программ – это Serial Monitor (монитор последовательного порта), встроенный в IDE. Цель данного раздела – рассмотреть концепции программирования на языке C через разработку простых программ, а затем загрузки и запуска их на плате. Читатели, хорошо владеющие языком C и знакомые с Serial Monitor, могут пропустить этот раздел<sup>1</sup>.

### Пример 2.1. Сумма целых чисел

Напишите программу, которая считывает целое число  $N$  с клавиатуры, а затем вычисляет и отображает сумму всех целых чисел от 1 до  $N$ .

**Решение.** Ниже показан листинг программы под названием *sumN*. Комментарии в начале текста программы используются для описания функций программы. Кроме того, здесь указаны имена автора и программы, а также дата разработки. Автор настоятельно рекомендует включать подобные комментарии в ваши программы, чтобы было легко отслеживать назначение и функциональность программ, а также их возможные изменения в будущем.

Функция `setup()` выполняется только один раз в начале программы. В рамках этой функции монитор последовательного порта настраивается на работу со скоростью 9600 бод<sup>2</sup> (при желании вы можете выбрать другую скорость передачи данных).

Основная программа выполняется в функции `loop()`. Здесь переменные `i`, `N` и `Sum` объявлены как целые числа и `Sum` также сбрасывается в 0. Программа предлагает пользователю ввести  $N$ , которое считывается с помощью встроенной функции `parseInt()`. Затем проверяется наличие данных перед чтением с клавиатуры и формируется цикл `for`, в котором вычисляется сумма всех целых чисел от 1 до  $N$  и сохраняется в переменной `Sum`. Наконец, выводится сумма в виде целого числа с помощью функции `println()`. Обратите внимание, что функция `println()` выводит неотображаемые символы возврат каретки / перевод строки после отображения данных. Программа останавливается с помощью оператора `while(1)` в конце, иначе функция `loop()` будет повторяться бесконечно.

<sup>1</sup> Строго говоря, Arduino IDE использует не C, а некий гибрид C/C++ на основе компилятора AVRGCC, дополненный специфическими Arduino-функциями и приемами. Поэтому даже знатокам языка рекомендуется проработать этот раздел, если ранее им не приходилось иметь дело с Arduino IDE.

<sup>2</sup> В рамках Arduino принято говорить о скорости передачи данных последовательного порта, измеряемой в *бодах* (*baud*, символах в секунду). Строго говоря, в данном случае речь идет о скорости в битах в секунду (*bps*), которая может превышать скорость в бодах (в зависимости от способа кодирования), но последовательный порт UART в микроконтроллерах обычно использует простейшее двоичное кодирование, для которого боды и биты в секунду совпадают.

Последовательность тестирования программы следующая:

- подключите макетную плату Arduino Uno R4 к ПК и настройте последовательный порт;
- введите программу, показанную на рис. 2.6 (или загрузите из архива книги на веб-сайте Elektor по указанной выше ссылке), а затем сохраните ее с подходящим именем, например *sumN*;
- нажмите **Sketch** (*Скетч*) → **Verify/Compile** (*Проверить/компилировать*), чтобы скомпилировать программу. Статус компиляции будет отображаться на нижней панели надписью «Compiling sketch....» (*Компиляция эскиза...*). Если проявятся какие-либо ошибки, вам следует вернуться в программу, чтобы их исправить. Если ошибок нет, нижняя панель отобразится, как показано на рис. 2.6 (нажмите **Output** (*Вывод*) на нижней панели, чтобы увидеть это сообщение);

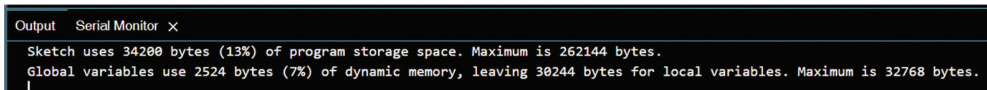


Рис. 2.6. Успешная компиляция

- нажмите **Sketch** (*Скетч*) → **Upload** (*Загрузить*), чтобы загрузить правильно скомпилированный код в процессор вашей макетной платы. В нижней части экрана вы должны увидеть сообщение «Done uploading» (*Загрузка завершена*);
- убедитесь, что для скорости передачи данных установлено значение 9600. Щелкните **Serial Monitor** в верхней части нижней панели (рис. 2.7). Если этот пункт монитора последовательного порта недоступен, щелкните значок **Serial Monitor** в правом верхнем углу экрана;

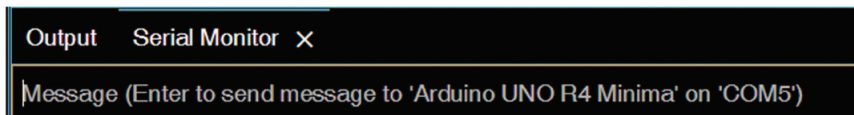


Рис. 2.7. Вызов монитора последовательного порта

### Листинг 2.1. Программа *sumN*

```
//-----
// SUM OF INTEGER NUMBERS FROM 1 to N
// =====
//
// This program calculates and displays the sum of integer numbers
//
// Author: Dogan Ibrahim
// File : sumN
// Date : June, 2023
//-----
```

```

void setup()
{
  Serial.begin(9600);
  delay(5000);
}
void loop()
{
  int i, N, Sum = 0;
  Serial.print("How many numbers are there?");
  while(Serial.available() <= 0);
  N = Serial.parseInt();
  Serial.print(N);
  for (i = 1; i <= N; i++)    // Do for 1 to N
  {
    Sum = Sum + i;          // Calculate the sum
  }
  Serial.println();         // Print new line
  Serial.print("Sum of numbers from 1 to "); // Display "Sum = "
  Serial.print(N);         // Display N
  Serial.print(" are: ");  // Display " are: "
  Serial.println(Sum);     // Display the sum
  while(1);                // Stop the program
}

```

- в окне монитора последовательного порта вы должны увидеть текст «How many numbers are there?» (*Сколько здесь чисел?*). Введите, например, 10 в строке, где написано: «Enter to send message to 'Arduino UNO R4' Minima...» (*Введите, чтобы отправить сообщение в «Arduino UNO R4» Minima...*), и нажмите клавишу <Enter> на своем ПК. Вы должны увидеть число 55, которое представляет собой сумму целых чисел от 1 до 10, как показано на рис. 2.8.

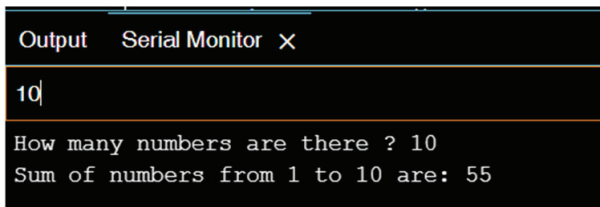


Рис. 2.8. Отображение суммы чисел от 1 до 10

Чтобы протестировать другие программы, представленные в следующих разделах этой главы, вам также следует выполнять действия, описанные выше.

### Пример 2.2. Таблица квадратов

Напишите программу для вывода таблицы квадратов целых чисел от 1 до 10.

**Решение.** Ниже показан листинг программы *squares*. Монитор последовательного порта инициализируется, как в предыдущем примере. В основном



цикле программы устанавливается цикл `for`, который выводит квадраты чисел от 1 до 10. Выводимые числа разделяются знаком табуляции («\t»).

### Листинг 2.2. Программа *squares*

```
//-----
// TABLE OF SQUARES
// =====
//
// This program displays table of squares of integers from 1 to 10
//
// Author: Dogan Ibrahim
// File : squares
// Date : June, 2023
//-----
void setup()
{
  Serial.begin(9600);
  delay(5000);
}
void loop()
{
  int i, N;
  Serial.println("TABLE OF SQUARES FROM 1 TO 10");
  Serial.println("=====");
  Serial.println("N\tSQARE");
  for (i = 1; i <= 10; i++) // Do for 1 to N
  {
    N = i * i; // Calculate the square
    Serial.print(i);
    Serial.print("\t");
    Serial.println(N);
  }
  while(1); // Stop the program
}
```

На рис. 2.9 показаны результаты работы программы, отображаемые на мониторе последовательного порта.

### Пример 2.3. Объем цилиндра

Напишите функцию, возвращающую объем цилиндра, где радиус и высота должны быть переданы в качестве аргументов функции. Используйте функцию в программе, чтобы вычислить и отобразить площадь цилиндра радиусом 10 см и высотой 15 см.

**Решение.** Объем цилиндра, радиус и высота которого равны  $r$  и  $h$  соответственно, определяется по формуле:

$$\text{Объем} = \pi \cdot r^2 \cdot h.$$

Ниже показан листинг программы *cylarea*. Функция `volume` получает радиус и высоту цилиндра в виде чисел с плавающей запятой и возвращает значение

объема в основную программу, которая затем отображает эту величину на мониторе последовательного порта.

TABLE OF SQUARES FROM 1 TO 10	
N	SQUARE
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

Рис. 2.9. Результаты программы *squares*

### Листинг 2.3. Программа *cylarea*

```
//-----
// VOLUME OF A CYLINDER
// =====
//
// This program calculates and displays the volume of a cylinder
//
// Author: Dogan Ibrahim
// File : cylarea
// Date : June, 2023
//-----
#define pi 3.14159
float r = 10.0;
float h = 15.0;
void setup()
{
  Serial.begin(9600);
  delay(5000);
}
//
// Function to calculate the volume
//
float volume(float radius, float height)
{
  float vol;
  vol = pi * radius * radius * height;
  return vol;
}
void loop()
```

```

{
  float cylinder;
  Serial.println("Volume of a cylinder with r = 10 cm and h = 15 cm");
  Serial.println("=====");
  cylinder = volume(r, h);
  Serial.print("Volume = ");
  Serial.print(cylinder);
  Serial.println(" cm3");
  while(1);          // Stop the program
}

```

На рис. 2.10 показан результат работы программы, при условии что в начале программы радиус и высота установлены на 10 см и 15 см соответственно.

```

Volume of a cylinder with r = 10 cm and h = 15 cm
=====
Volume = 4712.38 cm3

```

Рис. 2.10. Результат программы *cylarea*

### Пример 2.4. Перевод градусов Цельсия в градусы Фаренгейта

Напишите программу, которая будет получать температуру в градусах Цельсия, конвертировать ее в градусы Фаренгейта и затем отображать ее на мониторе последовательного порта. Программа должна считывать температуру с клавиатуры.

**Решение.** Температуру в градусах Цельсия  $C$  можно преобразовать в градусы Фаренгейта  $F$  по следующей формуле:

$$F = 1,8 \times C + 32.$$

Ниже показан листинг программы *CtoF*. Функция *ToF* получает в качестве аргумента градусы  $C$ , преобразует их в градусы  $F$  и возвращается в основную программу. Температура в градусах Цельсия считывается с клавиатуры.

#### Листинг 2.4. Программа *CtoF*

```

//-----
// CELSIUS TO FAHRENHEIT
// =====
//
// This program converts Celsius to Fahrenheit
//
// Author: Dogan Ibrahim
// File : CtoF
// Date : June, 2023
//-----
void setup()

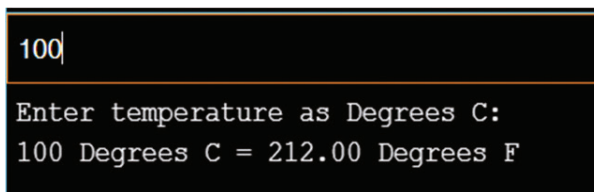
```

```

{
  Serial.begin(9600);
  delay(5000);
}
//
// Function to convert Degrees C to Degrees F
//
float ToF(float C)
{
  return (1.8 * C + 32);
}
void loop()
{
  float F;
  int C;
  Serial.println("Enter temperature as Degrees C: ");
  while(Serial.available() <= 0);
  C = Serial.parseInt();           // Read Degrees C
  F = ToF(C);                     // Convert to F
  Serial.print(C);
  Serial.print(" Degrees C = ");
  Serial.print(F);
  Serial.println(" Degrees F");
  while(1);                       // Stop the program
}

```

На рис. 2.11 показан результат работы программы, где 100 °C преобразуются в градусы Фаренгейта и отображаются на мониторе последовательного порта.



```

100
Enter temperature as Degrees C:
100 Degrees C = 212.00 Degrees F

```

Рис. 2.11. Результат программы CtoF

### Пример 2.5. Таблица умножения

Напишите программу для чтения целого числа, а затем составьте таблицу умножения от 1 до 12 для данного числа.

**Решение.** Ниже показан листинг программы *times*. Целое число считывается с клавиатуры и сохраняется в переменной *N*. Затем устанавливается цикл *for*, работающий от 1 до 12. Число *N* в этом цикле умножается на числа от 1 до 12 и затем отображается на мониторе последовательного порта.

**Листинг 2.5.** Программа *times*

```
//-----
// TIMES TABLE
// =====
//
// This program reads an integer number and then tabulates the time table
//
// Author: Dogan Ibrahim
// File : times
// Date : June, 2023
//-----
void setup()
{
  Serial.begin(9600);
  delay(5000);
}
void loop()
{
  int N, i;
  Serial.print("Enter the number: "); // Prompt for a number
  while(Serial.available() <= 0);
  N = Serial.parseInt();           // Read a number
  Serial.println(N);              // Display the number
  Serial.print("\nTimes table for number "); // Heading
  Serial.println(N);
  for(i = 1; i <=12; i++)         // Do 12 times
  {
    Serial.print(i);             // Display 1..12
    Serial.print(" X ");        // Display " X "
    Serial.print(N);            // Display N
    Serial.print(" = ");        // Display " = "
    Serial.println(i * N);      // Display the product
  }
  while(1);                      // Stop the program
}
}
```

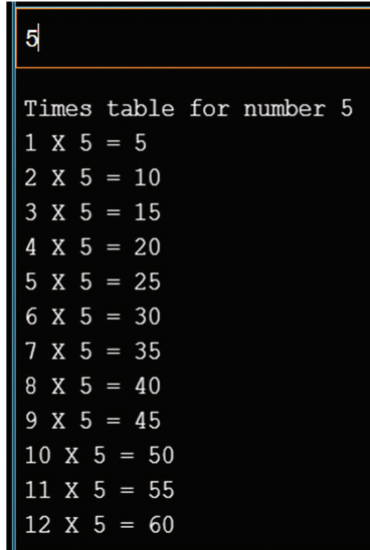
На рис. 2.12 показаны результаты программы, где отображается таблица умножения для числа 5.

**Пример 2.6.** Таблица синуса

Напишите программу вывода тригонометрической таблицы синуса для значений углов от  $0^\circ$  до  $90^\circ$  с шагом  $5^\circ$ .

**Решение.** Ниже показан листинг программы *sines*. После отображения заголовка запускается цикл `for`, который выполняется для значений от 0 до 90 с шагом 5. В этом цикле подсчитывается синус заданных углов.

Обратите внимание, что углы для тригонометрических функций необходимо вводить в радианах. Градусы преобразуются в радианы путем умножения на число  $\pi/180$ .



```

5
Times table for number 5
1 X 5 = 5
2 X 5 = 10
3 X 5 = 15
4 X 5 = 20
5 X 5 = 25
6 X 5 = 30
7 X 5 = 35
8 X 5 = 40
9 X 5 = 45
10 X 5 = 50
11 X 5 = 55
12 X 5 = 60

```

Рис. 2.12. Результаты программы *times***Листинг 2.6.** Программа *sines*

```

//-----
// TRIGONOMETRIC SINE
// =====
//
// This program displays trigonometric sine from 0 to 90 degrees
//
// Author: Dogan Ibrahim
// File : sines
// Date : June, 2023
//-----
#define pi 3.14159
void setup()
{
  Serial.begin(9600);
  delay(5000);
}
void loop()
{
  int degree;
  float rad;
  Serial.println("TRIGONOMETRIC SINE"); // Heading
  Serial.println("=====");
  Serial.println("Degree\tSine");
  for(degree = 0; degree <=90; degree += 5) // 0 to 90
  {
    Serial.print(degree); // Display degree
    Serial.print("\t"); // Tab
    rad = degree * pi / 180.0; // Convert to radians

```

```

Serial.println(sin(rad));           // Display sine
}
while(1);                           // Stop the program
}

```

На рис. 2.13 показан результат работы программы.

TRIGONOMETRIC SINE	
Degree	Sine
0	0.00
5	0.09
10	0.17
15	0.26
20	0.34
25	0.42
30	0.50
35	0.57
40	0.64
45	0.71
50	0.77
55	0.82
60	0.87
65	0.91
70	0.94
75	0.97
80	0.98
85	1.00
90	1.00

Рис. 2.13. Результат программы *sines*

### Пример 2.7. Таблица синуса, косинуса и тангенса

Напишите программу вывода тригонометрической таблицы синуса, косинуса и тангенса для значений углов от  $0^\circ$  до  $45^\circ$  с шагом  $5^\circ$ .

**Решение.** Программа аналогична приведенной выше, но здесь также включены косинус и тангенс. Ниже показан листинг программы *trigs*.

#### Листинг 2.7. Программа *trigs*

```

//-----
// TRIGONOMETRIC SINE,COSINE,TANGENT
// =====
//
// This program displays trigonometric sine,cosine,tangent

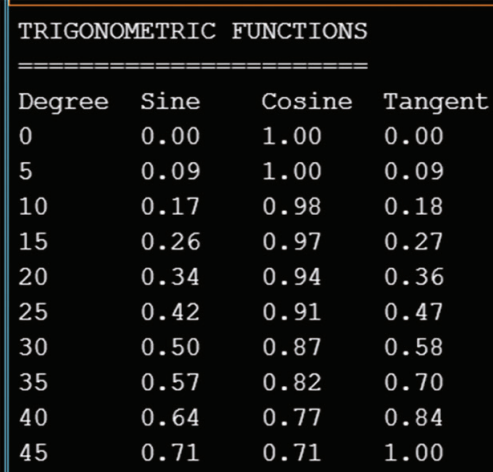
```

```

//
// Author: Dogan Ibrahim
// File : trigs
// Date : June, 2023
//-----
#define pi 3.14159
void setup()
{
  Serial.begin(9600);
  delay(5000);
}
void loop()
{
  int degree;
  float rad;
  Serial.println("TRIGONOMETRIC FUNCTIONS"); // Heading
  Serial.println("=====");
  Serial.println("Degree\tSine\tCosine\tTangent");
  for(degree = 0; degree <=45; degree += 5) // 0 to 45
  {
    Serial.print(degree);           // Display degree
    Serial.print("\t");             // Tab
    rad = degree * pi / 180.0;      // Convert to radians
    Serial.print(sin(rad));         // Display sine
    Serial.print("\t");             // Tab
    Serial.print(cos(rad));        // Display cosine
    Serial.print("\t");             // Tab
    Serial.println(tan(rad));      // Display tangent
  }
  while(1);                        // Stop the program
}

```

На рис. 2.14 показаны результаты работы программы.



TRIGONOMETRIC FUNCTIONS			
=====			
Degree	Sine	Cosine	Tangent
0	0.00	1.00	0.00
5	0.09	1.00	0.09
10	0.17	0.98	0.18
15	0.26	0.97	0.27
20	0.34	0.94	0.36
25	0.42	0.91	0.47
30	0.50	0.87	0.58
35	0.57	0.82	0.70
40	0.64	0.77	0.84
45	0.71	0.71	1.00

Рис. 2.14. Результаты программы trigs