

Использование Markdown, RMarkdown и bookdown для подготовки научных отчетов и дипломных работ

Н.О. Стрелков, В.В. Крутских

2023-10-03

Содержание

Аннотация	4
Введение	5
1 Теоретические основы использования Markdown	7
1.1 Язык разметки Markdown	7
1.2 Синтаксис Markdown	8
1.2.1 Заголовки	9
1.2.2 Эффекты шрифта	10
1.2.3 Списки	11
1.2.4 Ссылки и сноски	12
1.2.5 Изображения	12
1.2.6 Таблицы	13
1.2.7 Формулы	15
1.2.7.1 Формулы с матрицами	16
1.2.7.2 Формулы с интегралами и дифференциалами . .	17
1.2.7.3 Примеры формул	18
1.2.8 Список литературы и библиографические ссылки	20
1.2.9 Листинги исходного кода	20
1.2.10 Оглавление	21
1.2.11 Вычисления RMarkdown	22
1.3 Необходимое программное обеспечение	22
1.3.1 R, RStudio, TeXLive	22
1.3.1.1 Установка под Windows	22
1.3.1.2 Установка под GNU/Linux	24
1.3.1.2.1 Установка под Debian	24
1.3.1.2.2 Установка под ALT Linux	25
1.3.2 Текстовый редактор ReText	26
1.4 Процесс подготовки odt-документов	26
1.4.1 Использование odt-шаблона	27
1.4.2 Финальная подготовка рукописи	27
1.5 Процесс подготовки docx-документов	28
1.5.1 Использование docx-шаблона	28
1.5.2 Преобразование формул	28

1.5.3	Финальная подготовка рукописи	29
2	Вторая глава	31
3	Третья глава	32
	Заключение	33
	Приложения	34
	Список использованных источников	35

Аннотация

В настоящей работе рассматривается использование языка разметки Markdown для написания отчетов и дипломных работ. Используется расширение языка RMarkdown с дополнением bookdown. Демонстрируется применимость bookdown для оформления научных работ. Этот документ может быть использован как шаблон для подготовки текста работы.

Введение

В настоящее время для написания дипломов, отчетов и научных работ широко используется Microsoft Word (далее MS Word). Следует отметить, что MS Word имеет следующие существенные недостатки:

- форматы doc и docx - бинарные (не текстовые), поэтому сложно сравнивать разные версии документа между собой и искать текст;
- сложно организовать совместную работу;
- сложно автоматизировать нумерацию разделов, рисунков, таблиц, формул, сносок, списка литературы, списка используемых сокращений, иллюстраций, алфавитных указателей в больших документах (книги, дипломы, диссертации);
- не удобно вставлять ссылки на картинки, расположенные в отдельных файлах;
- не удобно вставлять код программ, расположенных в отдельных файлах;
- медленная и ненадежная работа с большими документами;
- формулы сохраняются в MathType (бинарный формат, не позволяет сравнивать версии между собой, только визуально);
- сложность создания и работы с многофайловыми документами (с мастер-документом).

Но MS Word обладает следующими достоинствами:

- широкое распространение программы;
- возможность создать документ практически любой сложности и произвольного формата.

Альтернативами для MS Word могут выступать следующие программы:

- использование бинарных форматов
 - LibreOffice Writer - по смыслу тоже самое, что и MS Word;
 - WPS Office - по смыслу тоже самое, что и MS Word;
 - Google Docs - удобная совместная работа, но нет автоматической нумерации объектов и прочего.
- текстовые форматы

- LaTeX - отличный язык для подготовки высококачественных документов любого объема и сложности, но имеет слишком перегруженный синтаксис и высокий порог вхождения;
- Легковесные языки разметки (Lightweight Markup Language) на основе простого текста (plain-text):
 - * Markdown - 2004 г. - <http://daringfireball.net/projects/markdown>;
 - * reStructuredText - 2002 г. - <http://docutils.sourceforge.net/rst.html>;
 - * AsciiDoc - 2013 г. - <http://asciidoc.org/>;
 - * MediaWiki - 2002 г. - <https://www.mediawiki.org/>;
 - * Emacs Org-Mode - 2003 г. - <https://orgmode.org/>.

Настоящий документ подготовлен с помощью легковесного языка разметки Markdown. Рассмотрим использование этого языка более подробно. В последующих главах будут рассмотрены примеры использования Markdown, RMarkdown и bookdown.

Глава 1

Теоретические основы использования Markdown

1.1 Язык разметки Markdown

Язык разметки Markdown создал Джон Грубер совместно с Аароном Шварцем в 2004 году. Ниже представлена полная цитата, отражающая цель создания языка:

John Gruber created the Markdown language in 2004 in collaboration with Aaron Swartz on the syntax, with the goal of enabling people “to write using an easy-to-read, easy-to-write plain text format, and optionally convert it to structurally valid XHTML (or HTML)”.

Официальный логотип языка Markdown представлен на рисунке



В настоящее время язык широко используется следующими сайтами и организациями:

- GitHub.com - GitHub Flavored Markdown (GFM) - сайт с публичными и частными Git-репозиториями;
- BitBucket.org - сайт с публичными и частными Git- и Mercurial- репозиториями;
- GitLab.com - система управления проектами с поддержкой Git-репозиторияев;
- Atlassian JIRA - система управления задачами и проектами;
- WordPress.com - веб-платформа для создания сайтов и блогов;
- StackOverflow.com - сеть сайтов вопросов и ответов по множеству тематик.

Язык Markdown обладает следующими преимуществами:

- текстовый удобный для редактирования и чтения формат;
- легко просматривать отличия между версиями и искать текст файловым менеджером и даже командой `grep`;

- возможна совместная работа в системе контроля версий (Git, Mercurial и пр.).

Для редактирования и просмотра Markdown документов может использоваться текстовый редактор с поддержкой Markdown:

- ReText (Windows, Linux - <https://github.com/retext-project/retext>);
- GhostWriter (Windows, Linux - <https://wereturtle.github.io/ghostwriter>);
- Typora (Windows, Linux, macOS - <https://typora.io/>);
- Remarkable (Windows, Linux - <http://remarkableapp.github.io/>);
- Geany (Windows, Linux, macOS - <https://geany.org/>);
- Atom text editor (Windows, Linux, macOS - <https://atom.io/>);
- Dillinger (online - <https://dillinger.io>).

Практически все редакторы имеют одинаковые стандартные возможности: форматирование, отображение Markdown и экспорт в HTML, PDF, ODT (OpenDocument) и др.

Настоящий документ написан с использованием обобщенного синтаксиса Markdown, поэтому изучение синтаксиса языка можно выполнять путем просмотра и/или редактирования отдельных элементов текста документа.

1.2 Синтаксис Markdown

Далее рассматривается обобщенный Markdown синтаксис, включающий в себя:

- исходный **Markdown** синтаксис (см. <http://daringfireball.net/projects/markdown>);
- расширение **RMarkdown** (см. RMarkdown Cheat Sheet и RMarkdown Reference);
- **bookdown** для написания книг (см. книгу Yihui Xie “bookdown: Authoring Books and Technical Documents with R Markdown”).

Исходный синтаксис **Markdown** обеспечивает форматирование документа. **RMarkdown** позволяет выполнять расчеты на языке программирования



R (см. <https://www.r-project.org/>) и оформлять полученные результаты в одном документе.

Bookdown – это расширение RMarkdown для создания книг, его создал Yihui Xie в 2016 году. Официальная книга постоянно обновляется и расположена по адресу <https://bookdown.org/yihui/bookdown/>.

Назначение bookdown: автоматизированное получение из одного Rmd-документа файлов любых форматов (PDF, epub, HTML, docx, Mobi).

Дополнительная функциональность bookdown:

- Автоматическое создание оглавления.
- Автоматическая одноуровневая или двухуровневая нумерация рисунков, таблиц, формул, теорем, доказательств и т.п.
- Построение списка литературы в тексте учетом выбранного стиля оформления (например, в порядке упоминания источников).
- Использование вычислительных возможностей языка R - возможность расчета графиков и вставки их в текст, включение файлов в текст.
- Возможность выполнения программ скриптов в операционной системе и их включения в текст.
- Автоматическое создание алфавитного указателя в LaTeX и PDF.

Программное обеспечение и особенности. Для работы с RMarkdown



требуется установить среду разработки RStudio и TeXLive для экспорта в LaTeX и PDF. Эти вопросы рассмотрены подробнее в разделе 1.3. При этом процесс получения документов имеет вид:



Выходными форматами являются PDF, ODT, RTF, HTML, docx. Подробности этого процесса описаны в книгах Yihui Xie.

Перейдем к рассмотрению обобщенного синтаксиса Markdown.

1.2.1 Заголовки

Заголовки в Markdown обозначаются с помощью символа “решетка” (#). Заголовок 1-го уровня имеет один символ #, заголовок 2-го уровня - “##” и т.п.

Заголовки могут иметь идентификатор, он указывается в фигурных скобках, см. например заголовок этого пункта. Все идентификаторы не должны содержать подчеркиваний (“_”), но могут содержать знаки минус (“-”).

Ссылка на нумерованный раздел с известным идентификатором будет иметь вид: см. раздел 1.2.1.

Так же ссылка может быть задана с произвольным текстом: см. введение. Этот вариант предпочтителен для экспорта в DOCX.

1.2.2 Эффекты шрифта

Абзацы текста отделяется друг от друга переводами строки до и после.

Перевод текста на новую строку выполняется с помощью добавления двух пробелов

в конце строки (здесь три слова “в конце строки” оказались вначале новой строки).

Полужирный шрифт может быть получен одним из способов: **полужирный** или **полужирный**.

Курсивный шрифт может быть получен одним из способов: *курсив* или *курсив*.

Полужирный курсив может быть получен одним из способов: ***полужирный курсив*** или ***полужирный курсив***.

После этого абзаца следует горизонтальная линия, заданная в виде трех последовательных знаков минус (“-”):

Моноширинный шрифт (обычно используется для отображения исходного кода) может быть задан с помощью двух символов машинописного обратного апострофа - код в тексте.

Многострочный исходный код отделяется от основного текста четырьмя пробелами:

```
int main() {  
return 0;  
}
```

Цитаты или примечания оформляются с помощью знака больше (“>”):

Это цитата или примечание.

Поскольку настоящий документ компилируется в RStudio, поэтому здесь поддерживаются дополнительные возможности RMarkdown:

- верхний индекс, например для возведения в квадрат - x^2 ;
- нижний индекс для индексации элементов - y_3 ;
- зачеркнутых текст с помощью двойного знака тильды (“~”) : - ;
- знак тире (“-”) в виде двух последовательных знаков минус : понятие — определение ;
- знак длинного тире (“—”) в виде трех последовательных знаков минус : понятие — понятие .

1.2.3 Списки

Ниже представлен нумерованный список из трех пунктов (нумерация выполняется автоматически)

1. Первый элемент списка;
2. Второй элемент списка;
3. Третий элемент списка.

Примечание: автоматическая нумерация вложенных списков не поддерживается.

Ниже представлен ненумерованный список из трех элементов

- Верхний элемент списка;
- Средний элемент списка;
- Нижний элемент списка.

Списки могут быть вложенными, при этом уровни отделяются двумя пробелами:

- 1-й элемент уровня 1
- 2-й элемент уровня 1
 - 1-й элемент уровня 2
 - * 1-й элемент уровня 3

Элементы списка могут содержать форматирование (полужирный шрифт, курсив, моноширинный шрифт и т.п.), могут содержать вложенные элементы. При этом код должен быть отделен необходимым дополнительным количеством пробелов:

- строка 1
- строка 2 с блоком кода

```
int main() {  
    return 0;  
}
```

Далее следует цитата

Цитата

и нумерованный список

1. один
 2. два
 3. три
- строка 3

1.2.4 Ссылки и сноски

Ниже представлена **ссылка** на сайт университета:

Сайт НИУ “МЭИ”

В квадратных скобках указан текст ссылки, который будет отображаться на экране, а в круглых скобках указывается полный URL ресурса.

В этом абзаце имеется **сноска**¹ с расшифровкой в конце страницы.

1.2.5 Изображения

Простой Markdown не позволяет выполнять автоматическую нумерацию рисунков, но позволяет вставить **рисунок** с подписью. Далее следует рисунок с названием *Markdown*, сохраненный в файле **figures/Markdown-mark.png**:



Рис. 1.1: Markdown

Рисунок может не иметь названия, тогда код упрощается:



Автоматическая нумерация рисунков выполняется с помощью метки `fig`, поэтому рисунок с автоматической нумерацией и идентификатором `md-logo` может быть вставлен следующим образом:



Рис. 1.2: Логотип Markdown

Ссылка на такой рисунок будет иметь вид: на рисунке 1.2 представлен логотип языка разметки Markdown.

Текст, обозначающий иллюстрацию (например, “Рис.” или “Рисунок”) задается в файле `_bookdown.yml` в YAML-секции `language`:

```
language:
  label:
    fig: 'Рисунок '
```

Для каждого рисунка могут быть принудительно заданы его размеры с помощью соответствующих атрибутов `width` или `height` внутри фигурных

¹Текст сноски

скобок после описания изображения. Поддерживаются следующие единицы измерения `px`, `cm`, `mm`, `in / inch` и `%`. Для сохранения пропорций рекомендуется задавать один размер (ширину или высоту).

Пример задания ширины изображения:



Рис. 1.3: Логотип Markdown шириной 1 см

Пример задания высоты изображения:



Рис. 1.4: Логотип Markdown высотой 2 см

Пример задания ширины и высоты изображения:



Рис. 1.5: Логотип Markdown шириной 64 мм и высотой 39 мм

1.2.6 Таблицы

Таблицы могут оформляться одним из способов - первый:

Столбец 1	Столбец 2
строка 1, столбец 1	строка 1, столбец 2
строка 2, столбец 1	строка 2, столбец 2

второй:

Столбец 1	Столбец 2
строка 1, столбец 1	строка 1, столбец 2
строка 2, столбец 1	строка 2, столбец 2

Выравнивание столбцов выполняется с помощью знака двоеточия (":") во второй строке:

Столбец 1 (влево)	Столбец 2 (по центру)	Столбец 3 (вправо)
строка 1, столбец 1	строка 1, столбец 2	строка 1, столбец 3
строка 2, столбец 1	строка 2, столбец 2	строка 2, столбец 3

Автоматическая нумерация таблиц выполняется с помощью метки `tab` аналогично рисункам.

Таблица 1.4: Технические характеристики платы Arduino Uno R3

Параметр	Значение
Микроконтроллер	ATmega328P
Напряжение питания	5 В
Внешнее напряжение питания (рекомендуемое)	7–12 В
Внешнее напряжение питания (предельное)	6–20 В

Примечание: английская надпись “Table:” обязательно должна присутствовать в начале строки для правильной нумерации таблицы.

Ссылка на таблицу выполняется аналогично рисунку: в таблице 1.4 представлены технические характеристики платформы Arduino Uno.

Текст, обозначающий таблицу (например, “Табл.” или “Таблица”) задается в файле `_bookdown.yml` в YAML-секции `language`:

```
language:
  label:
    tab: 'Таблица '
```

Текст в таблицах может иметь форматирование.

В некоторых случаях требуется поместить несколько строк текста в одну ячейку таблицы. При этом используется другой способ задания таблиц, который показан ниже при оформлении таблицы 1.5.

Таблица 1.5: Программы и языки программирования для математических расчетов

Визуальные	Текстовые
Платные • PTC MathCAD • Wolfram Mathematica • Waterloo Maple Бесплатные • SMath Studio • WolframAlpha • wxMaxima	• MathWorks MATLAB • Scilab • GNU Octave • Sage • FreeMAT • Axiom • Reduce • Euler Math Toolbox • Julia • Spyder - The Scientific PYthon Development EnviRonment • R, RStudio • написание программ на языках Fortran, C/C++ с библиотеками LAPACK, BLAS, ATLAS, MINPACK, NETLIB.org, Armadillo

Примечание: еще большее количество типов таблиц можно найти внутри документа проекта GostDown или в соответствующем разделе документации по Pandoc.

1.2.7 Формулы

Формулы без нумерации и расположенные непосредственно могут быть набраны в одиночных знаках доллара, например, $a^2 + b^2 = c^2$.

Формулы без нумерации и расположенные в отдельной строке могут быть набраны в двойных знаках доллара в отдельной строке, как показано ниже

$$E = \frac{mc^2}{\sqrt{1 - \frac{v^2}{c^2}}}$$

или

$$E = \frac{mc^2}{\sqrt{1 - \frac{v^2}{c^2}}}$$

Формулы с автоматической нумерацией отличаются наличием идентификатора в круглых скобках ($\backslash \#eq:name$):

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (1.1)$$

При необходимости формула и ее номер могут быть правильно выравнены в MS Word с помощью применения пользовательского стиля *DisplayEquation*:

$$S_{AM}(t) = U(1 + M \cdot F(t)) \cos(\omega_c \cdot t) \quad (1.2)$$

Ссылка на формулу выполняется с помощью поля $\backslash @ref(eq:name)$: ссылаемся на первое выражение - (1.1) и второе выражение (1.2).

Набор формул возможен в MathType 6 (желательно 6.9 и выше) с последующим экспортом в LaTeX с помощью меню *Preferences - Cut and Copy Preferences*: в списке *MathML or TeX* нужно выбрать пункт *Plain TeX*.

Для быстроты копирования можно снять обе галочки: *Include translator name in translation* и *Include MathType data in translation*.

Но иногда при экспорте в docx возникают ошибки pandoc вида [pandoc warning] Cannot convert the following TeX math, skipping:, поэтому лучше использовать LyX для набора или исправления формул.

1.2.7.1 Формулы с матрицами

Матрицы:

$$AA = \begin{pmatrix} \alpha_{22} & \beta \\ \Xi^2 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}, \left\| \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \right\|$$

Матрица с точками:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Система уравнений со скобкой:

$$\begin{cases} 2|x|(2-x) = a, \\ x < 2, \\ x \neq 0. \end{cases}$$

1.2.7.2 Формулы с интегралами и дифференциалами

В этом разделе собраны символы, наиболее часто используемые в дифференциальном и интегральном исчислении:

- `lim` предел;
- `prod` произведение;
- `sum` сумма;
- `frac` черта деления;
- `int` интеграл;
- `iint` двойной интеграл;
- `iiint` тройной интеграл;
- `oint` круговой интеграл;
- `partial` частная производная;
- `infty` бесконечность;
- `to` стрелка (в пределах);
- `pm` плюс-минус

Дроби, в которых числитель расположен над знаменателем, набираются с помощью команды `\frac`.

Производная в виде дроби:

$$F(x) = 2 + \frac{d}{dx}(U(x)) \quad (1.3)$$

Частная производная:

$$dz = \frac{\partial z}{\partial x}dx + \frac{\partial z}{\partial y}dy \quad (1.4)$$

Интегралы:

$$\int_0^3 f(x)dx \quad (1.5)$$

$$\iint_{x^2+y^2=1} f(x,y)dxdy \quad (1.6)$$

$$\iiint_{x^2+y^2+z^2=1} f(x,y,z)dxdydz \quad (1.7)$$

Пределы:

$$\lim_{x \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e \quad (1.8)$$

Произведение:

$$F(x) = \prod_{n=1}^{\infty} \left(1 + \frac{x}{n!}\right) \quad (1.9)$$

Функции:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (1.10)$$

$$\arg, \cos, \cosh, \cot, \coth, \csc, \det, \dim, \exp, \gcd, \hom, \inf, \ker, \lg, \ln \quad (1.11)$$

$$\log, \max, \min, \sec, \sin, \sinh, \sup, \tan, \tanh, \arccos, \arcsin, \arctan \quad (1.12)$$

$$\hat{\Phi}[k, l] = \begin{cases} 0 & \text{if } k, l = 0 \\ S_x[k, l] \cdot H_x[k, l] + S_y[k, l] \cdot H_y[k, l] & \text{otherwise} \end{cases} \quad (1.13)$$

$$S_{\text{вых}}(x_2, y_2) = A_0 \underbrace{\iint dx_0 dy_0 g(x_0, y_0) \cdot h(x_2 - x_0, y_2 - y_0)}_{\text{по определению это есть свёртка}} = A_0 g \otimes h \quad (1.14)$$

1.2.7.3 Примеры формул

Амплитудная модуляция:

$$S_{AM}(t) = U(1 + M \cdot F(t)) \cos(\omega t + \phi) \quad (1.15)$$

Амплитудная однотоновая модуляция:

$$S_{AM}(t) = U(1 + M \cdot \cos(\Omega t)) \cos(\omega_c t + \phi) \quad (1.16)$$

Амплитудная однотоновая модуляция, разложение:

$$\begin{aligned} S_{AM}(t) &= U(1 + M \cdot \cos(\Omega t)) \cos(\omega_c t) = \\ &= U \cos(\omega_c t) + U \cdot M \cdot \cos(\Omega t) \cos(\omega_c t) = \\ &= U \cos(\omega_c t) + \frac{U \cdot M}{2} \cos((\omega_c - \Omega)t) + \frac{U \cdot M}{2} \cos((\omega_c + \Omega)t) \end{aligned} \quad (1.17)$$

$$\begin{aligned}
S_{AM}(t) = & \underbrace{U \cos(\omega_c t)}_{\text{несущая}} + \\
& + \underbrace{\frac{U \cdot M}{2} \cos((\omega_c - \Omega)t)}_{\text{нижняя боковая полоса}} + \\
& + \underbrace{\frac{U \cdot M}{2} \cos((\omega_c + \Omega)t)}_{\text{верхняя боковая полоса}}
\end{aligned}$$

Частотная модуляция:

$$S_{FM}(t) = U \cos((\omega_c + D \cdot F(t))t + \phi) \quad (1.18)$$

Фазовая модуляция:

$$S_{PM}(t) = U \cos(\omega_c t + (\phi + D \cdot F(t))) \quad (1.19)$$

Коэффициент передачи ФНЧ:

$$K(\omega) = \frac{1}{1 + j\omega\tau} \quad (1.20)$$

Коэффициент передачи ФВЧ:

$$K(\omega) = \frac{j\omega\tau}{1 + j\omega\tau} \quad (1.21)$$

Последовательный колебательный контур:

$$K(\omega) = \frac{U_C}{E} = \frac{\frac{1}{j\omega C}}{j\omega L + \frac{1}{j\omega C} + R} \quad (1.22)$$

$$K(\omega) = \frac{1}{1 - \omega^2 LC + j\omega RC} \quad (1.23)$$

Формула с прямым русским (кириллическим) текстом:

$$U_{\text{вых.}} = U_{\text{м}} \cos(\omega t) \quad (1.24)$$

Формула с курсивным русским (кириллическим) текстом:

$$U_{\text{вых.}} = U_{\text{м}} \cos(\omega t) \quad (1.25)$$

1.2.8 Список литературы и библиографические ссылки

Список литературы задается в виде отдельного файла с записями в формате BibTeX (*.bib*). Пример фрагмента файла *bibliography.bib*:

```
@online{arduino,  
  title = {Arduino - Home},  
  url = {https://www.arduino.cc/},  
}  
  
@book{banzi,  
  title = {Arduino для начинающих волшебников},  
  publisher = {М.: Рид Групп},  
  author = {{Массимо Банци}},  
  year = {2012},  
}
```

Описание библиографической записи в формате BibTeX для книг может быть получено с сайта Google Books - нужно найти книгу с помощью поиска, найти ссылку *About this book* и перейти по ней, далее на открывшейся странице с описанием книги найти в области *Export Citation* кнопку *BiBTeX* и скопировать описание книги в *bib*-файл.

Каждый элемент *bib*-файла характеризуется типом (в этом примере это *@online* для веб-страницы и *@book* для книги) и идентификатором, следующим после открывающей фигурной скобки (в этом примере *arduino* и *banzi*).

Ссылка в тексте дается в квадратных скобках с указанием идентификатора объекта после символа *@*.

Пример ссылки в тексте: для изучения платформы Arduino см. веб-страницу проекта Arduino [1] и книгу М. Банци [2].

При необходимости можно добавить ссылку с учетом номеров страниц: см. книгу М. Банци [2, с. 11] или [2, с. 15–20].

Оформление списков литературы выполняется в соответствии с правилами, заданными в CSL-файле стиля ссылок в YAML-преамбуле. К этому документу подключен файл *gost-r-7-0-5-2008-numeric.csl*, соответствующий ГОСТ Р 7.0.5-2008 с цифровой нумерацией источников в порядке их упоминания в тексте.

1.2.9 Листинги исходного кода

В некоторых случаях оказывается необходимым выполнять нумерацию фрагментов исходного кода (листингов). Для этого в файле *_bookdown.yml* в секции *language* можно переобозначить индекс, предназначенный для оформления примеров:

```
language:
  label:
    exm: 'Листинг '
```

В этом случае фрагмент программы может быть оформлен с использованием пользовательского стиля *ListingCaption* для docx следующим образом:

Листинг 1.1. Программа для отображения работы программы с помощью светодиода

```
// зададим номер контакта, к которому подключен светодиод
int ledPin = 13;

// функция setup выполняется однократно при нажатии клавиши
void setup() {
  // инициализация контакта, к которому подключен светодиод
  pinMode(ledPin, OUTPUT);
}

// функция loop выполняется в бесконечном цикле
void loop() {
  digitalWrite(ledPin, HIGH); // включаем светодиод (HIGH
это напряжение 5 В)
  delay(1000);                // пауза 1000 мс
  digitalWrite(ledPin, LOW);  // выключаем светодиод (в эт
это напряжение 0 В)
  delay(1000);                // пауза 1000 мс
}
```

Ссылка на листинг исходного кода оформляется в виде: см. листинг 1.1.

1.2.10 Оглавление

Оглавление или содержание задается в YAML-преамбуле документа и имеет вид:

```
---
toc-title: "Оглавление"
output:
  bookdown::word_document2:
    toc: true
    toc_depth: 5
  bookdown::html_document2:
    toc: true
    toc_depth: 5
---
```

здесь `toc-title` задает текст, выводимый перед оглавлением, `toc: true` включает отображение оглавления, `toc_depth` задает количество отображаемых уровней заголовков в оглавлении.

1.2.11 Вычисления RMarkdown

RMarkdown позволяет объединять в одном документе расчеты и оформление результатов. Ниже представлен код для создания вложенного нумерованного списка:

1. Пункт
 - 1.1. Подпункт
 - 1.2. Подпункт
2. Пункт

1.3 Необходимое программное обеспечение

Для локальной работы необходимо установить следующие компоненты: язык программирования R, среду RStudio, и набор типографских программ TeXLive для подготовки LaTeX- и PDF-версий документа и редактор ReText для редактирования документов Markdown.

Для редактирования полученного docx-файла необходим Microsoft Office 2007 (с SP3 и со всеми обновлениями) или более новой версии.

Для работы с формулами внутри docx-файла необходим MathType версии 6.9 и выше.

1.3.1 R, RStudio, TeXLive

1.3.1.1 Установка под Windows

Поддерживается 64-битная версия Windows 7 и выше. Процесс установки сводится к следующим шагам:

1. Загрузить R 4.0.5 for Windows с официального сайта и установить со всеми настройками по умолчанию.
2. Загрузить R for Windows Build Tools с официального сайта и установить со всеми настройками по умолчанию.
3. Загрузить редактор Notepad++ с официального сайта и установить со всеми настройками по умолчанию.

4. Для сборки демонстрационного примера потребуется загрузить Git for Windows с официального сайта и установить, выбрав на этапе *Choosing the default editor used by Git* пункт *Use Notepad++ as Git's default editor*.
5. Для удобства работы с Git-репозиториями рекомендуется дополнительно загрузить с официального сайта и установить расширение TortoiseGit для Проводника.
6. Загрузить RStudio с официального сайта и запустить установку файла RStudio-....exe и дождаться ее завершения.
7. Для сборки PDF-версии документа с помощью RStudio потребуется установить TeX-пакеты от проекта TeXLive.

Примечание: в сети попадаются инструкции по использованию под Windows дистрибутива MikTeX, но он не работает нормально совместно с RStudio.

В дистрибутивах GNU/Linux применяется TeXLive, поэтому будем использовать именно его и под Windows.

Установка TeXLive требует следующих действий:

1. Загрузить дистрибутив TeXLive в виде ISO-файла как торрент с официального сайта.
 2. Подключить загруженный образ установочного диска в систему программой OSFMount или аналогичной.
 3. Запустить установщик `install-tl-windows.bat`.
 4. В открывшемся окне нажать кнопку *Установить*.
 5. По окончании установки нажать кнопку *Заккрыть*.
8. Запустить RStudio и установить пакеты R для поддержки bookdown через командную строку внутри RStudio, выполнив последовательно команду в окне *Console*:

```
install.packages(c('markdown', 'bookdown', 'tikzDevice',  
project.org'))
```

и закрыть RStudio.

9. Для подготовки первого RMarkdown документа следует клонировать демонстрационный русифицированный репозиторий с помощью Git - открыть Проводник, нажать правую кнопку мыши, выбрать в выпадающем меню пункт *Git Bash Here*, вставить в командную строку команду

```
git clone https://github.com/k0ly4n/bookdown-demo-rus
```

нажать <Enter> и дождаться завершения клонирования, закрыть окно *MINGW....*

10. Перейти в только что созданный каталог `bookdown-demo-rus` и выполнить двойной щелчок мышью по файлу `bookdown-demo-rus.Rproj` для открытия проекта книги в RStudio.
11. Дождаться открытия RStudio, выбрать в верхней правой части окна вкладку *Build* и нажать стрелку вниз рядом с пунктом *Build Book* и выбрать пункт *bookdown:html_document2* для подготовки HTML-версии книги.

Примечание: возможно появление окна *Install Required Packages*, в котором нужно нажать кнопку *Yes* для установки пакетов.

По завершению сборки откроется HTML-версия книги.

Примечание: если требуется подготовка книги в EPUB-формате (*bookdown:epub_book*), то для просмотра этого формата нужно установить дополнительный просмотрщик. Например, простой и легковесный Sumatra PDF.

12. Для сборки PDF-версии документа следует выбрать в верхней правой части окна RStudio вкладку *Build* и нажать стрелку вниз рядом с пунктом *Build Book* и выбрать пункт *bookdown:pdf_book*.

1.3.1.2 Установка под GNU/Linux

В настоящее время существует два скрипта автоматизированной установки необходимого программного обеспечения:

1. `umpris` предназначен для систем на основе дистрибутива Debian, он поддерживает системы Debian (9, 10, 11 и 12), AstraLinux (2.12 и 1.7), Ubuntu MATE (18.04 LTS, 20.04 LTS и 22.04 LTS).
2. `alpis` - для систем ALT Linux, созданных на основе 9-й и 10-й платформ (Workstation, MATE StarterKit и SimplyLinux).

Многолетний опыт автора настоящего документа позволяет рекомендовать среду рабочего стола MATE для повседневного и повсеместного использования. В следующем подразделе приведены в том числе команды для установки среды MATE в систему Debian.

1.3.1.2.1 Установка под Debian В системах Debian рекомендуется предварительно настроить использование команды `sudo` с помощью выполнения следующих команд, запущенных от имени пользователя *root*:


```
apt-get update
apt-get install sudo
```

и затем добавить пользователя в группу *sudo* командой `su -l -c "usermod -a -G sudo $USER"`, выполненной от обычного пользователя, и перезагрузить компьютер.

После этого среда MATE может быть установлена с помощью команды `sudo apt-get install task-mate-desktop mate-desktop-environment-extras`.

Для запуска скрипта `umpis.sh` в графической сессии MATE следует выполнить следующие команды в терминале:

```
sudo apt-get update
sudo apt-get install wget dconf-cli
```

```
cd ~/Downloads || cd ~/Загрузки
wget -c https://raw.githubusercontent.com/N0rbert/umpis/main/umpis.sh
chmod +x umpis.sh
sudo -E ./umpis.sh
```

Большую часть времени скрипт работает автономно без участия пользователя. На финальном этапе скрипт задаст вопрос о выборе экрана входа в систему, на который нужно ответить выбором пункта *sddm*; запросит у пользователя его пароль для установки среды программирования Arduino IDE и путь к каталогу ее установки. После окончания работы скрипта нужно перезагрузить компьютер, на экране входа выбрать сеанс MATE и авторизоваться.

1.3.1.2.2 Установка под ALT Linux В системах ALT Linux Workstation и MATE StarterKit рекомендуется предварительно настроить использование команды `sudo` с помощью выполнения следующей команды, запущенной от обычного пользователя:

```
su -l -c "usermod -a -G wheel $USER; control sudo wheelonly;
```

Для запуска скрипта `alpis.sh` в графической сессии MATE следует выполнить следующие команды в терминале:

```
sudo apt-get update
sudo apt-get install wget lsb-release
```

```
cd ~/Downloads || cd ~/Загрузки
wget -c https://raw.githubusercontent.com/N0rbert/alpis/main/alpis.sh
chmod +x alpis.sh
sudo -E ./alpis.sh
```

Скрипт работает автономно. После окончания его работы нужно перезагрузить компьютер, на экране входа выбрать сеанс MATE и авторизоваться.

1.3.2 Текстовый редактор ReText

Редактирование отдельных Markdown файлов можно выполнять в простом визуальном редакторе ReText. В зависимости от версии Windows его установка выполняется следующим образом:

- в Windows 7 нужно установить Python 3.8.10 (поставив галочку *Add Python 3.8 to PATH*) и затем установить ReText из репозитория PyPi с помощью команды `pip install ReText==7.2.3 pyqtwebengine==5.15.2`.
- в более новых версиях Windows нужно установить актуальную версию Python и затем установить ReText из репозитория PyPi с помощью команды `pip install ReText pyqt6-webengine`.

После установки редактор ReText можно запустить командой `retext` и создать затем ярлык для программы.

Примечание: в случае если при открытии документа с русским текстом символы в редакторе отображаются некорректно, следует изменить шрифты на поддерживающие кириллицу. Для этого следует нажать в меню **Edit** на пункт **Change editor font** для смены шрифта в редакторе и выбрать нужный шрифт (обычно моноширинный, например *Courier New*).

Далее следует повторить действия для шрифта предварительного просмотра **Change preview font** и выбрать нужный шрифт (обычно не моноширинный, например *Times New Roman*).

В системах GNU/Linux редактор содержится в репозиториях под именем пакета `retext`. Для поддержки проверки орфографии нужно установить специальный пакет командой `sudo apt-get install python3-enchant` (Debian) или `sudo apt-get install python3-module-enchant` (ALT Linux).

Следует учитывать, что редактор ReText не поддерживает расширения RMarkdown. Однако, это не мешает их использованию и набору в тексте.

Для поддержки формул внутри текста (вида $\$ \dots \$$) нужно включить в настройках расширение `mathjax` (в линуксе с помощью команды `echo mathjax > $HOME/.config/markdown-extensions.txt`) и переключить режим предварительного просмотра на использование движка *WebEngine (Chromium) Renderer* (или *WebKit Renderer*) в меню *Edit*.

Для игнорирования YAML заголовка нужно добавить в строку дополнений `meta`.

1.4 Процесс подготовки odt-документов

Для odt-документов требуются дополнительные действия.

1.4.1 Использование odt-шаблона

Для удовлетворения требований оформления под конкретный случай возможно создание и использование файла odt-шаблона. В этом файле задается формат страницы, поля страницы, стили заголовков и текста. Путь в файлу odt-шаблона задается в YAML-преамбуле:

```
---
output:
  bookdown::odt_document2:
    reference_docx: styles-reference.odt
---
```

Файл шаблона может быть многократно последовательно изменен до получения качественного результата.

Особенности:

1. На текущий момент не поддерживаются пользовательские стили оформления участков текста.
2. Формулы будут отображаться в формате OMMML, их преобразование в MathType под GNU/Linux невозможно.

1.4.2 Финальная подготовка рукописи

Финальная подготовка текста включает следующие этапы:

- оформление **страниц** и их порядок следования:

1. Редактирование титульного листа в соответствии с требованиями.
2. Перенос страницы задания и/или аннотации на второй лист (после титульного листа) перед оглавлением.
3. Удаление нумерации перед следующими структурными элементами: *Аннотация, Введение, Заключение, Приложения, Список использованных источников* (нужно нажать мышью на номер и выполнить удаление кнопкой <Delete>).
4. Ручной перенос приложений в конец документа (после списка использованных источников).
5. Ручное добавление заголовков “Продолжение Приложения #” в случае если приложения состоят двух и более страниц.

- оформление **таблиц**:

1. Перенос названий и номеров таблиц снизу таблицы наверх.
2. Ручное добавление текста “Продолжение табл. #” в случае если таблица не помещается на одну страницу.

- оформление **формул**:

1. Выравнивание формулы и ее номера вручную с помощью соответствующего стиля.

- действия **перед печатью**:

1. Проверка правописания с корректировкой ошибок.
2. Обновление всех ссылок документа с помощью последовательного нажатия комбинаций клавиш <Ctrl+A>, <F9> или из меню *Tools* ☐ *Update* ☐ *Update All*.

1.5 Процесс подготовки docx-документов

Аналогично формату odt, для docx-документов требуются дополнительные действия.

1.5.1 Использование docx-шаблона

Для удовлетворения требований оформления под конкретный случай возможно создание и использование файла docx-шаблона. В этом файле задается формат страницы, поля страницы, стили заголовков и текста. Путь в файлу docx-шаблона задается в YAML-преамбуле:

```
---
output:
  bookdown::word_document2:
    reference_docx: styles-reference.docx
---
```

Файл шаблона может быть многократно последовательно изменен до получения качественного результата.

1.5.2 Преобразование формул

RStudio использует Pandoc для преобразования документов из Markdown в другие форматы. Поэтому формулы оказываются преобразованными в формат OMML (Microsoft's Office Math Markup Language) вместо MathType. Но для издательства МЭИ и при оформлении дипломов и отчетов достаточно строго требуется использование MathType, поэтому необходимо выполнить конвертацию формул из OMML в MathType:

1. Открыть docx-документ в Word (иногда *Открыть и восстановить*);

2. На ленте *MathType* нажать кнопку *Convert Equations*; в открывшемся окне установить в области *Equation types to convert* галочку *Word 2007 and late (OMML) equations*, затем в области *Convert equations to* выбрать *MathType equations (OLE objects)*.
3. Нажать кнопку *Convert* и ждать результата.
4. Отформатировать все формулы (греческие буквы - прямо и т.п.) - нажать на ленте *MathType* кнопку *Format Equations*, в области *Format equations using preferences from* установить переключатель в положение *MathType preference file*, нажать кнопку *Browse*, выбрать файл настроек *format.eqp* и применить изменения кнопкой *OK*.

Примечание: При конвертации из OMML в MathType могут не работать LaTeX-окружения *split* и *multiline*, поэтому надежнее использовать *array*.

1.5.3 Финальная подготовка рукописи

Финальная подготовка текста включает следующие этапы:

- оформление **страниц** и их порядок следования:

1. Выделение первой страницы под титульный лист и ее добавление в начало документа.
2. Перенос страницы задания и/или аннотации на второй лист (после титульного листа) перед оглавлением.
3. Удаление нумерации перед следующими структурными элементами: *Аннотация, Введение, Заключение, Приложения, Список использованных источников* (нужно нажать мышью на номер и выполнить удаление кнопкой *<Delete>*).
4. Ручной перенос приложений в конец документа (после списка использованных источников).
5. Ручное добавление заголовков “Продолжение Приложения #” в случае если приложения состоят двух и более страниц.

- оформление **таблиц**:

1. Ручное добавление текста “Продолжение табл. #” в случае если таблица не помещается на одну страницу.

- оформление **формул**:

1. Преобразование всех формул из OMML в MathType (см. раздел 1.5.2 выше).

- действия **перед печатью**:

1. Проверка правописания с корректировкой ошибок.
2. Обновление всех ссылок документа с помощью последовательного нажатия комбинаций клавиш <Ctrl+A>, <F9> и выбора пункта *Обновить целиком* в диалоговом окне.
3. Проверка отсутствия неработоспособных ссылок с помощью поиска (<Ctrl+F>, текст *Ошибка!*).

Глава 2

Вторая глава

Это – вторая глава

Глава 3

Третья глава

Это – третья глава

Заключение

В заключении оратор сказал...

Приложения

Ну, это у кого не вырвали.
Мы про... аппендикс, если что...

Список использованных источников

В том числе и целебных...

1. Arduino – Home [Электронный ресурс]. URL: <https://www.arduino.cc/>.
2. Массимо Банци. Arduino для начинающих волшебников. М.: Рид Групп, 2012.