

1. Введение

Класс *pcbdos* предоставляет разработчикам печатных плат возможность вёрстки конструкторской документации с помощью юникодной версии текстового процессора \LaTeX --- \XeTeX . Подразумевается, что пользователь имеет понятие о \TeX и способен сверстать в \LaTeX простейший документ. С помощью *pcbdos* возможна вёрстка перечней элементов, спецификаций, чертежей печатных плат, сборочных чертежей и схем электрических принципиальных. Для вёрстки перечней элементов и спецификаций высокая квалификация пользователя \LaTeX не требуется. Не требуется она и для вёрстки чертежей печатных плат, сборочных чертежей и схем электрических принципиальных, если рисунки, полученные из системы разработки печатных плат, не требуют доработки. В этом случае необходимо просто вставить заранее подготовленный рисунок в тело документа *pcbdos*. В противном случае требуется знакомство пользователя с пакетом *tikz*.

Данный класс был написан для личных целей, когда понадобилось оформить конструкторскую документацию на уже разработанные печатные платы и сдать твёрдые копии документов в архив. При разработке *pcbdos* ставилась задача получения таких выходных документов, которые позволили бы беспрепятственно пройти нормоконтроль на конкретном предприятии. Хотя полное соответствие ЕСКД и не являлось главной целью *pcbdos*, *сознательные* нарушения стандартов при разработке класса не допускались. Автор надеется, что данный класс пригодится кому-то ещё. В случае, если Вы найдёте *pcbdos* полезным, но в чём-то не соответствующим ЕСКД или не позволяющим пройти Ваш нормоконтроль, пожалуйста, сообщите об этом автору. Возможно, вместе мы исправим это недоразумение.

2. Установка

Класс *pcbdos* разрабатывался и использовался на компьютере с операционной системой *Debian GNU/Linux*. Вероятнее всего, он также будет работать и на *Windows*, и на *MacOS*.

На машине пользователя *pcbdos* должен быть установлен и настроен дистрибутив *TeX Live*. Процесс установки и настройки *TeX Live* для конкретной платформы описан в его официальной документации.

Для установки *pcbdos* нужно скопировать дерево исходных файлов класса в директорию, на которую указывает переменная `TEXMFHOME`, и установить используемые в *pcbdos* шрифты для конкретной операционной системы. `TEXMFHOME` является переменной дистрибутива *TeX Live*, указывающей на дерево, которое пользователи *TeX Live* могут использовать для установки собственных пакетов, шрифтов или обновлённых версий системных пакетов. Оно находится в домашней директории, своей для каждого пользователя.

Архив с классом *pcbdoc* содержит две директории, *texmf* и *.fonts*. В директории *texmf* находится директория *tex*, дерево исходных файлов *pcbdoc*. В директории *.fonts* находятся файлы используемых в *pcbdoc* шрифтов.

Далее подразумевается, что на машине пользователя установлена операционная система *Debian GNU/Linux*, а командной оболочкой является *bash*.

Переменная `TEXMFHOME` указывает на директорию *texmf* в домашнем каталоге пользователя. В этом можно убедиться, выполнив команду

```
tlmgr conf | grep TEXMFHOME
```

Таким образом, установка *pcbdoc* сводится к извлечению содержимого архива в домашний каталог пользователя:

```
unzip pcbdoc.zip -d
```

3. Тренинг

Вася нарисовал схему и разработал плату замечательного устройства --- фильтра низкой частоты первого порядка. Плата была успешно изготовлена, спаяна и настроена. Теперь это устройство безукоризненно выполняет свои функции на радость клиентам фирмы, где работает Вася. Да вот беда --- начальство Васи требует оформить перечень элементов согласно ЕСКД. Немного подумав, Вася решает использовать для этих целей *pcbdoc*.

Вася открывает свой любимый текстовый редактор(кто знает, возможно это *etacs*) и приступает к работе. Он аккуратно набирает первую строчку:

```
\documentclass[doctype=pe]{pcbdoc}
```

Теперь \LaTeX в курсе, что ему предстоит сверстать перечень элементов. Далее Васе требуется указать, что он является автором этого замечательного документа:

```
\AuthorSet{Пупкин}
```

Готовый перечень элементов Вася планирует показать для проверки коллеге, который, как и Вася, разрабатывает замечательные электронные устройства, рабочее место которого находится неподалёку. Ну в самом деле, а вдруг в документ закрадётся ошибка? Одна голова хорошо, а две лучше, думает Вася, набирая следующую строчку:

```
\CheckerSet{Ближайший}
```

Вася знает, что проверять готовый документ также будет и очень суровый нормоконтролёр. Нужно указать и этот факт. Быстро и решительно Вася набирает очередную строчку:

```
\NormControllerSet{Суровый}
```

Вася знает, что утверждать перечень элементов будет Васин начальник. Вася очень любит своего начальника, потому что он обещал повысить Васе зарплату. Жизнь прекрасна, думает Вася, набирая следующую строчку:

```
\ApproverSet{Сказочник}
```

Далее Вася указывает имя своего устройства и его десятичный номер, который по секрету сообщил Васе его начальник отдела:

```
\NameSet{Фильтр}  
\NumberSet{РОГА.123456.001}
```

Пора приступать к заполнению тела документа, думает Вася, набирая следующие две строчки:

```
\begin{document}  
\begin{ElementList}
```

Несмотря на то, что конденсатор в Васином устройстве всего один, Вася, вспомнив про суровость нормоконтролёра, решил создать для одного конденсатора целый раздел:

```
\Part{Конденсаторы}
```

Вот он, этот единственный конденсатор:

```
\Element{X7R_0805_1_МКФ_5%}{\refbox{C1}}{1}
```

Как видим, Вася не забыл указать его позиционное обозначение и количество. Аналогично Вася поступил и с единственным резистором:

```
\Part{Резисторы}  
\Element{RMC_0805_1_КОМ_5%}{\refbox{R1}}{1}
```

А вот соединителей в Васином устройстве целых два. Вася не забыл и про них:

```
\Part{Соединители}  
\Element{Розетка SMA-BJ}{\refbox{XS1,XS2}}{2}
```

Ну вроде всё, подумал Вася. Пора закругляться.

```
\end{ElementList}  
\end{document}
```

И сохранил полученный файл, присвоив ему имя *vasia.tex*.

Открыв эмулятор терминала, Вася перешёл в директорию, куда он сохранил свой файл, и скомпилировал его:

```
cd vasia  
xelatex vasia.tex
```

После компиляции Вася обнаружил в своей рабочей директории файл ***vasia.pdf***. Ну надо же, подумал Вася, получилось. Но что это за странные вопросительные знаки в графе ***Листов***? Наверное, я что-то не то сделал, подумал Вася, повторно запуская компиляцию:

```
xelatex vasia.tex
```

После которой странные знаки исчезли. То-то же, подумал Вася и приступил к настройке принтера. Распечатав полученный документ,

Поз. обозначение	Наименование	Кол.	Примечание
<i>Конденсаторы</i>			
C1	X7R_0805_1МКФ_5%	1	
<i>Резисторы</i>			
R1	RMC_0805_1КОМ_5%	1	
<i>Соединители</i>			
XS1, XS2	Разетка SMA-BJ	2	
<i>Подп. и дата</i>			
<i>Инф. № эддл</i>			
<i>Взам. инф. №</i>			
<i>Подп. и дата</i>			
РОГА.123456.001ПЗ			
Изм.	Лист	№ докум.	Подпись Дата
Разраб.	Пупкин		
Проб.	Ближайший		
Н. контр.	Сурабый		
Утв.	Сказочник		
Фильтр		Лит.	Лист Листов
Перечень элементов			1

Вася направился собирать подписи.

4. Опции и ключи класса `pcbdoc`

Опцией класса `pcbdoc` является идентификатор, влияющий на параметры вёрстки документа. Ключом класса `pcbdoc` является опция, имеющая некоторое значение. Ключ представляет собой конструкцию типа `<name>=<value>`, где `<name>` является

именем ключа, а *<value>* -- его значением. Ключи и опции указываются в необязательном аргументе команды `\documentclass` и отделяются друг от друга запятыми.

4.1. Тип документа

Ключ `doctype` определяет тип документа:

<code>pe</code>	Перечень элементов
<code>pp</code>	Чертёж печатной платы
<code>sb</code>	Сборочный чертёж
<code>sch</code>	Схема электрическая принципиальная
<code>spec</code>	Спецификация

Если ключ `doctype` не указан, типом документа по умолчанию является спецификация.

4.2. Размер страницы

Если типом документа является чертёж печатной платы, сборочный чертёж или схема электрическая принципиальная, с помощью ключа `papersize` имеется возможность указать размер страницы. Если же типом документа является перечень элементов или спецификация, ключ `papersize` игнорируется, и размер страницы устанавливается в значение по умолчанию. Если ключ `papersize` не указан, размер страницы также устанавливается в значение по умолчанию.

Ключ `papersize` может принимать значения `a4`, `a3`, `a2`, `a1`, `a4x3` и `a4x4`. Размером страницы по умолчанию является `a4`.

4.3. Толщины линий

Ключи `linethick` и `linethin` задают ширину толстой и тонкой линии соответственно. Значениями по умолчанию данных ключей являются `0.6mm` и `0.3mm` соответственно.

4.4. Лист изменений

Если типом документа является спецификация, опция `changelist` задаёт печать листа изменений в конце документа.

4.5. Прочие опции и ключи

По умолчанию `pcbdoc` печатает пустую строку после каждой записи в перечне элементов. При указании опции `compactmode` вышеуказанная пустая строка подавляется.

Опция **draftmode** может использоваться при вёрстке сборочного чертежа, схемы электрической принципиальной или чертежа печатной платы в *черновом режиме*. В черновом режиме вёрстки изменяется цвет фона и наносится координатная сетка, что облегчает применение специфичных для данного типа документа команд. С помощью ключа **gridstep** в этом случае имеется возможность указать шаг сетки. Ключ **gridstep** по умолчанию имеет значение **10mm**.

5. Шрифты

По умолчанию в *pcbdoc* используется наклонный шрифт **GOST type A**, размер которого зависит от контекста. В качестве прямого имеется возможность использовать шрифт **GOST Type AU**. Локально сменить наклонный шрифт на прямой и(или) изменить размер шрифта можно с помощью команд, приведённых в *Таблице 1* и *Таблице 2*.

Таблица 1. Команды изменения размера прямого шрифта

Команда	Пример использования	Результат
<code>\small</code>	<code>\AuthorSet{\small{}}Пупкин</code>	Пупкин
<code>\normalfont</code>	<code>\AuthorSet{\normalfont{}}Пупкин</code>	Пупкин
<code>\llarge</code>	<code>\AuthorSet{\llarge{}}Пупкин</code>	Пупкин
<code>\large</code>	<code>\AuthorSet{\large{}}Пупкин</code>	Пупкин
<code>\LARGE</code>	<code>\AuthorSet{\LARGE{}}Пупкин</code>	Пупкин
<code>\Large</code>	<code>\AuthorSet{\Large{}}Пупкин</code>	Пупкин

Таблица 2. Команды изменения размера наклонного шрифта

Команда	Пример использования	Результат
<code>\smallit</code>	<code>\AuthorSet{\smallit{}}Пупкин</code>	Пупкин
<code>\normalfontit</code>	<code>\AuthorSet{\normalfontit{}}Пупкин</code>	Пупкин
<code>\llargeit</code>	<code>\AuthorSet{\llargeit{}}Пупкин</code>	Пупкин
<code>\largeit</code>	<code>\AuthorSet{\largeit{}}Пупкин</code>	Пупкин
<code>\LARGEit</code>	<code>\AuthorSet{\LARGEit{}}Пупкин</code>	Пупкин
<code>\Largeit</code>	<code>\AuthorSet{\Largeit{}}Пупкин</code>	Пупкин

Кроме того, для указания типа шрифта и его размера можно воспользоваться встроенными средствами \LaTeX . Например:

```
\AuthorSet{\fontspec[Scale=0.68]{GOST type A}\itshape{}}Пупкин
```

6. Команды заполнения полей документа

Команды заполнения полей, приведённые в **Таблице 3**, должны находиться либо непосредственно в преамбуле документа, либо в отдельном пакете, включаемом в преамбулу с помощью команды `\usepackage`.

Таблица 3. Команды заполнения полей документа

Команда	Описание
<code>\AuthorSet{<name>}</code>	Печатает аргумент <code><name></code> в поле Разраб. основной надписи.
<code>\CheckerSet{<name>}</code>	Печатает аргумент <code><name></code> в поле Пров. основной надписи.
<code>\ScaleSet{<value>}</code>	Печатает аргумент <code><value></code> в поле Масштаб основной надписи сборочного чертежа или чертежа печатной платы.
<code>\NormControllerSet{<name>}</code>	Печатает аргумент <code><name></code> в поле Н. контр. основной надписи.
<code>\TechControllerSet{<name>}</code>	Печатает аргумент <code><name></code> в поле Т. контр. основной надписи сборочного чертежа или чертежа печатной платы.
<code>\ApproverSet{<name>}</code>	Печатает аргумент <code><name></code> в поле Утв. основной надписи.
<code>\NameSet{<name>}</code>	Печатает аргумент <code><name></code> в поле наименования изделия основной надписи. Аргумент <code><name></code> может быть как однострочным, так и двустрочным. Разделение аргумента на строки производится с помощью команды <code>\\</code> . Например: <code>\NameSet{Модуль\\расширителя сознания}</code>
<code>\NumberSet{<number>}</code>	Печатает десятичный номер <code><number></code> в поле обозначения документа основной надписи схемы электрической принципиальной, перечня элементов, сборочного чертежа и спецификации, а также в поле Перв. примен. схемы электрической принципиальной, перечня элементов, чертежа печатной платы и сборочного чертежа.
<code>\PcbNumberSet{<number>}</code>	Печатает десятичный номер <code><number></code> в поле обозначения документа основной надписи чертежа печатной платы.
<code>\PcbMaterialSet{<name>}</code>	Печатает аргумент <code><name></code> в поле обозначения материала детали основной надписи чертежа печатной платы. Аргумент <code><name></code> может быть однострочным, двустрочным или трёхстрочным. Разделение аргумента на строки производится с помощью команды <code>\\</code> . Например: <code>\PcbMaterialSet{Материал фольгированный\\качественный\\от надёжного поставщика}</code>
<code>\PrimaryUseSet{<number>}</code>	Печатает аргумент <code><number></code> в поле Перв. примен. основной надписи спецификации.

7. Перечень элементов

Исходный код перечня элементов, как и любого другого документа \LaTeX , должен начинаться с команды `\documentclass`. В основном аргументе команды (в фигурных скобках) следует указать класс документа *pcbdoc*, а в необязательном (в квадратных скобках) --- ключ *doctype=pe*.

```
\documentclass[doctype=pe]{pcbdoc}
```

Далее (в преамбуле) должны находиться команды заполнения полей документа. Их можно разместить либо непосредственно в преамбуле, либо в отдельном пакете, включаемом в преамбулу с помощью команды `\usepackage`. Тело документа, находящееся после преамбулы, должно начинаться с команды

```
\begin{document}
```

а заканчиваться командой

```
\end{document}
```

Внутри тела документа должно находиться окружение *ElementList*, которое начинается с команды

```
\begin{ElementList}
```

а заканчивается командой

```
\end{ElementList}
```

Внутри окружения *ElementList* должны находиться команды заполнения строк перечня элементов, приведённые в Таблице 4.

Таблица 4. Команды заполнения строк перечня элементов

Команда	Описание
<code>\Part{<name>}</code>	Печатает подчёркнутый аргумент <i><name></i> в центре колонки <i>Наименование</i> Например: <code>\Part{Микросхемы}</code>

Таблица 4. Команды заполнения строк перечня элементов. Продолжение

Команда	Описание
<code>\Element[<note>]</code> <code>{<naming>}</code> <code>{<refdes1 ... refdesN>}</code> <code>{<quantity>}</code>	<p>Заполняет строку перечня элементов. Необязательный аргумент <note> печатается в колонке Примечание. Аргументы <naming>, <refdes1 ... refdesN> и <quantity> печатаются в колонках Наименование, Поз. обозначение и Кол. соответственно. Каждая запись в аргументе <refdes1 ... refdesN> печатается в отдельной строке колонки Поз. обозначение. Каждую строку в колонке Поз. обозначение следует отделять от предыдущей одним или несколькими пробельными символами (символ перехода на другую строку также является пробельным символом). Между символом { и первой строкой, а также между последней строкой и символом } пробельных символов быть не должно, поэтому в примере ниже используется символ подавления последующих пробельных символов %. Каждую запись в строке позиционных обозначений рекомендуется размещать в аргументе команды <code>\refbox</code>, которая центрирует запись внутри колонки. Шрифт любой записи строки позиционных обозначений можно немного уменьшить с помощью команды <code>\llargeit</code>, что позволяет разместить в колонке чуть выступающий за её пределы текст. Например:</p> <pre> \Element{Y5V_1206_4,7_МКФ_20%_25V}{% \refbox{C6,C15,C16} \refbox{C18,C21} \refbox{C174,C175} \refbox{C180,C181} \refbox{C184-C187} \refbox{C190,C191} \refbox{C195-C199} \refbox{C201,C204} \refbox{C205} \refbox{\llargeit{C207-C210} \refbox{C212} \refbox{\llargeit{C232-C234} \refbox{\llargeit{C238-C240} \refbox{\llargeit{C265-C266}% }{36} </pre>
<code>\\</code>	Переход на новую строку
<code>\newpage</code>	Переход на новую страницу

8. Спецификация

Содержание

1	Введение	1
2	Установка	1
3	Тренинг	2
4	Опции и ключи класса <code>psbdos</code>	5
4.1	Тип документа	6
4.2	Размер страницы	6
4.3	Толщины линий	6
4.4	Лист изменений	6
4.5	Прочие опции и ключи	6
5	Шрифты	7
6	Команды заполнения полей документа	8
7	Перечень элементов	9
8	Спецификация	10