修订历史

|  |  |  |
| --- | --- | --- |
| 版本 | 日期 | 原因 |
| 0.9.0 | 2019/05/15 | 首版 |
|  |  |  |

目 录

[第1章 序言 1](#_Toc11148300)

[第2章 软件安装和使用 2](#_Toc11148301)

[2.1 LaTeX软件安装及简单配置 2](#_Toc11148302)

[2.1.1 软件安装 2](#_Toc11148303)

[2.1.2 软件使用环境简要配置 5](#_Toc11148304)

[2.1.3 LaTeX优缺点 6](#_Toc11148305)

[2.1.4 LaTeX命令和环境 7](#_Toc11148306)

[2.1.5 LaTeX源码结构 7](#_Toc11148307)

[2.1.6 文件类型说明 8](#_Toc11148308)

[2.1.7 文件组织方法 9](#_Toc11148309)

[第3章 手册命名及版本管理规范 10](#_Toc11148310)

[3.1 手册命名及版本管理规范 10](#_Toc11148311)

[3.1.1 芯片手册类别 10](#_Toc11148312)

[3.1.2 芯片手册命名规范 13](#_Toc11148313)

[3.1.3 版本管理及升序规则 14](#_Toc11148314)

[3.2 手册相关编码 14](#_Toc11148315)

[3.2.1 手册编码规则 14](#_Toc11148316)

[3.2.2 手册编码示例 16](#_Toc11148317)

[3.2.3 勘误信息编号规则 16](#_Toc11148318)

[第4章 手册编写规范 18](#_Toc11148319)

[4.1 手册LaTeX工程结构 18](#_Toc11148320)

[4.1.1 文件结构说明 18](#_Toc11148321)

[4.1.2 Make文件说明 18](#_Toc11148322)

[4.1.3 芯片手册工程生成 19](#_Toc11148323)

[4.1.4 章节源码及中间源码说明 21](#_Toc11148324)

[4.2 手册编写规范 22](#_Toc11148325)

[4.2.1 模板规范 22](#_Toc11148326)

[4.2.2 图片引用规范 23](#_Toc11148327)

[4.2.3 图片编辑规范 24](#_Toc11148328)

[4.2.4 表格编写及规范 25](#_Toc11148329)

[4.2.5 交叉引用方法及规范 26](#_Toc11148330)

[4.2.6 公式编写及规范 26](#_Toc11148331)

[4.2.7 历史修订记录填写规范 27](#_Toc11148332)

[4.3 LaTeX常用语法 27](#_Toc11148333)

[4.3.1 空格与分段 27](#_Toc11148334)

[4.3.2 注释 28](#_Toc11148335)

[4.3.3 特殊字符表示 28](#_Toc11148336)

[4.3.4 标点符号使用 28](#_Toc11148337)

[4.3.5 断行及断页 29](#_Toc11148338)

[4.3.6 章节和目录 30](#_Toc11148339)

[4.3.7 目录生成 30](#_Toc11148340)

[4.3.8 文档结构划分 30](#_Toc11148341)

[4.3.9 文本对齐 30](#_Toc11148342)

[4.3.10 交叉引用 31](#_Toc11148343)

[4.3.11 字体样式命令 31](#_Toc11148344)

[4.3.12 字号命令 32](#_Toc11148345)

[4.3.13 行距设置 33](#_Toc11148346)

[4.3.14 段落缩进格式 33](#_Toc11148347)

[4.3.15 垂直间距设置 34](#_Toc11148348)

[4.3.16 水平间距 34](#_Toc11148349)

[4.3.17 页边距设置 35](#_Toc11148350)

[4.3.18 页面内容垂直对齐 35](#_Toc11148351)

[4.3.19 使用颜色 36](#_Toc11148352)

[4.3.20 添加超链接 37](#_Toc11148353)

[4.4 LaTeX编译错误排查技巧 38](#_Toc11148354)

[4.4.1 报错排查示例 38](#_Toc11148355)

[4.4.2 处理方式 38](#_Toc11148356)

[4.4.3 常见的LaTeX错误及处理 38](#_Toc11148357)

[第5章 联合开发及迭代规则 41](#_Toc11148358)

[5.1 联合开发与迭代 41](#_Toc11148359)

[5.1.1 联合开发模式 41](#_Toc11148360)

[5.2 发布方式及规则 41](#_Toc11148361)

[附录A 附录 42](#_Toc11148362)

[A.1 常用宏 42](#_Toc11148363)

[A.1.1 颜色宏定义 42](#_Toc11148364)

[A.1.2 表格相关宏定义 42](#_Toc11148365)

[A.1.3 文档布局相关 42](#_Toc11148366)

[附录B 常用模版样式 44](#_Toc11148367)

[B.1 寄存器概述表 44](#_Toc11148368)

[B.2 寄存器描述表 45](#_Toc11148369)

# 序言

LaTeX是一种基于TEX的排版系统，由美国计算机学家莱斯利.兰伯特在20世纪80年代初期开发，利用这种格式，即使使用者没有排版和专业的程序设计知识，也可以充分发挥由TEX提供的强大功能，可以在短期内开发出具有书籍质量的印刷品。

目前我司与合作伙伴的贴牌芯片及合封芯片，均基于LaTeX工程进行编辑。为方便不同部门之间修订手册，尤其是部分SIP芯片，手册部分内容由专人撰写，出现多人维护同一份手册的情况，为了规范和提高手册的质量，简化编写手册的工作量，后续编写手册必须按照本文档规范进行，严控手册质量，修改后由专人审核，审核通过再进行合并及发布。

# 软件安装和使用

## LaTeX软件安装及简单配置

### 软件安装

1. 软件下载

Texlive基本上成为了TeX发行版本套装最为流行且更新最快的套装，这里推荐安装Texlive，下载链接为：<http://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/> ，根据自己的情况下载合适的版本，它是一款免费的软件。

1. 软件安装过程

以Texlive2018为例，打开上述链接，获取最新的软件 ，点击解压，如图2.1所示。

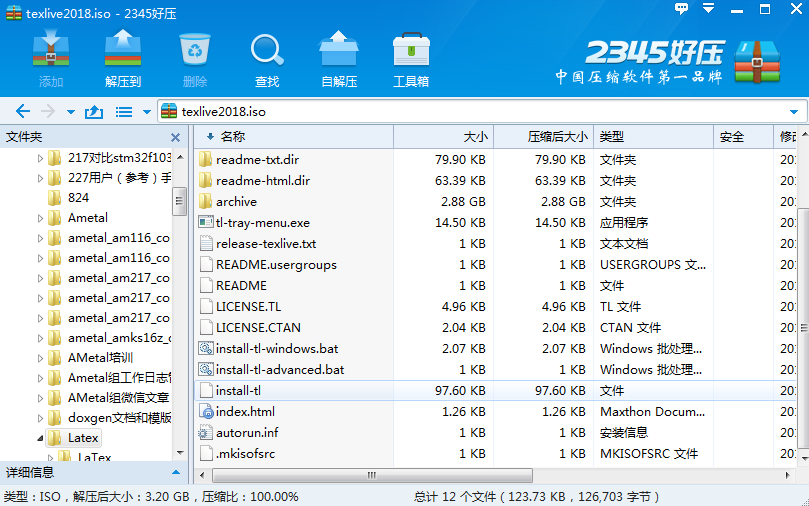


图2.1 解压Texlive

解压之后，点击如图2.2所示脚本进行安装。

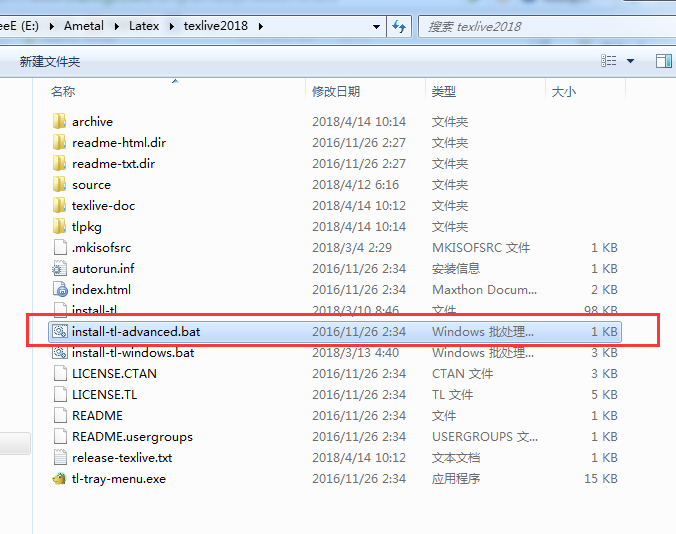


图2.2 点击安装脚本

按情况修改安装目录等路径信息，点击安装，如图2.3所示。

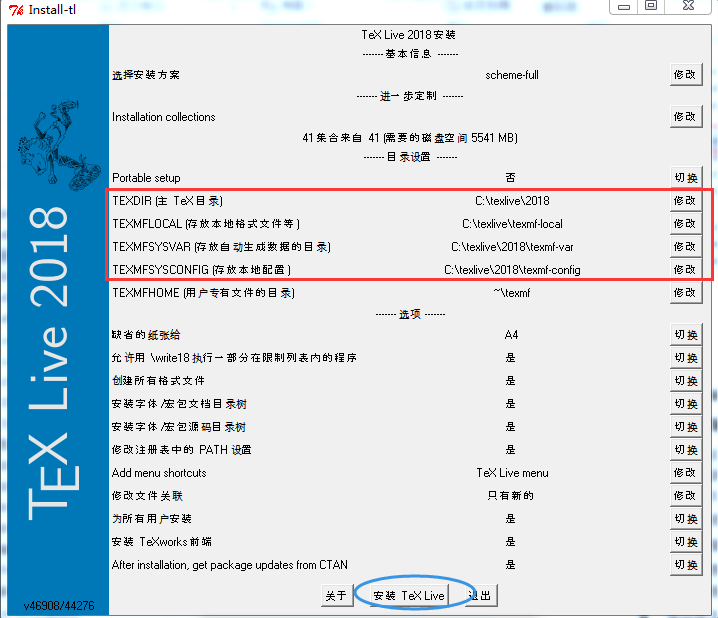


图2.3 修改路径

安装过程如下，时间较长，耐心等待，直到安装完成，如图2.4所示。

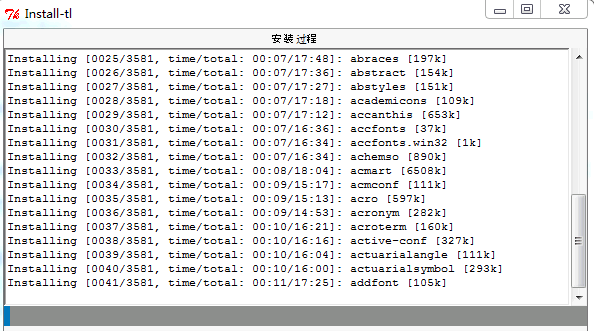


图2.4 安装过程

### 软件使用环境简要配置

软件安装完成之后，先进行简要配置，点击“选项”🡪 “设置”🡪 “常规”🡪 “语言”，选择“zh\_EN”，点击确认，配置为中文模式，方便使用，如图2.5所示。

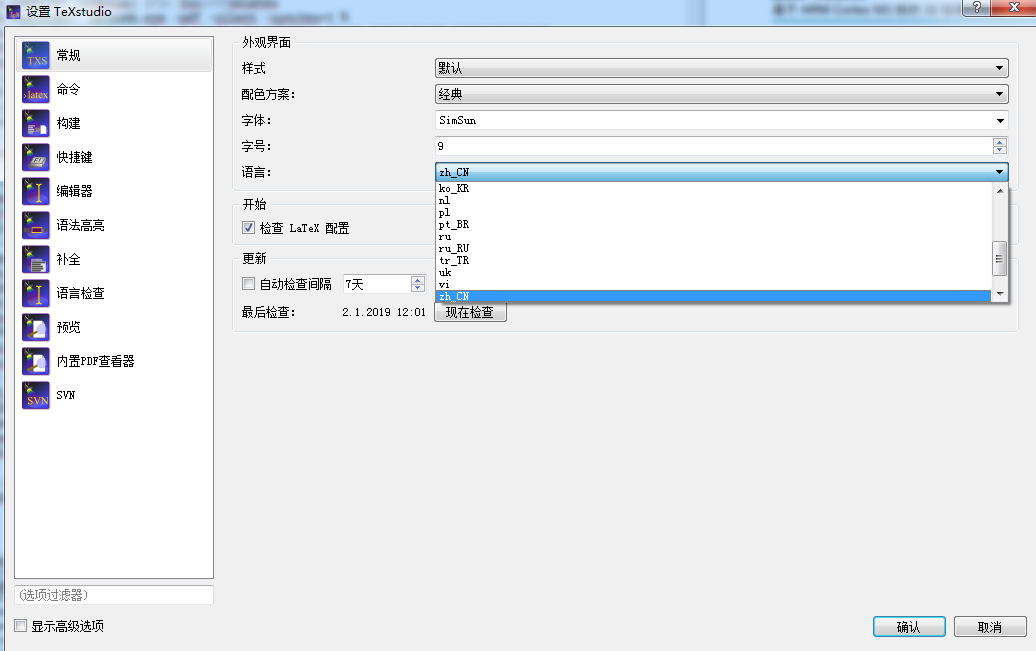


图2.5 汉化

编译器和文献工具设置，点击“选项”🡪 “设置”🡪 “构建”，如图2.6所示。

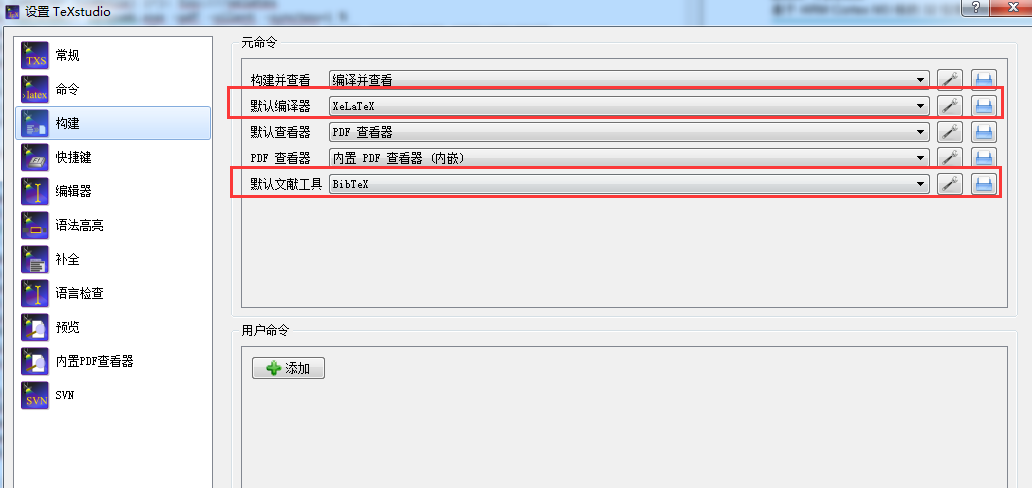


图2.6 编译器设置

打开工程，点击编译，编译完成之后，信息提示栏提示完成，如图2.7所示，编译完成之后，右侧为编译出的文档。

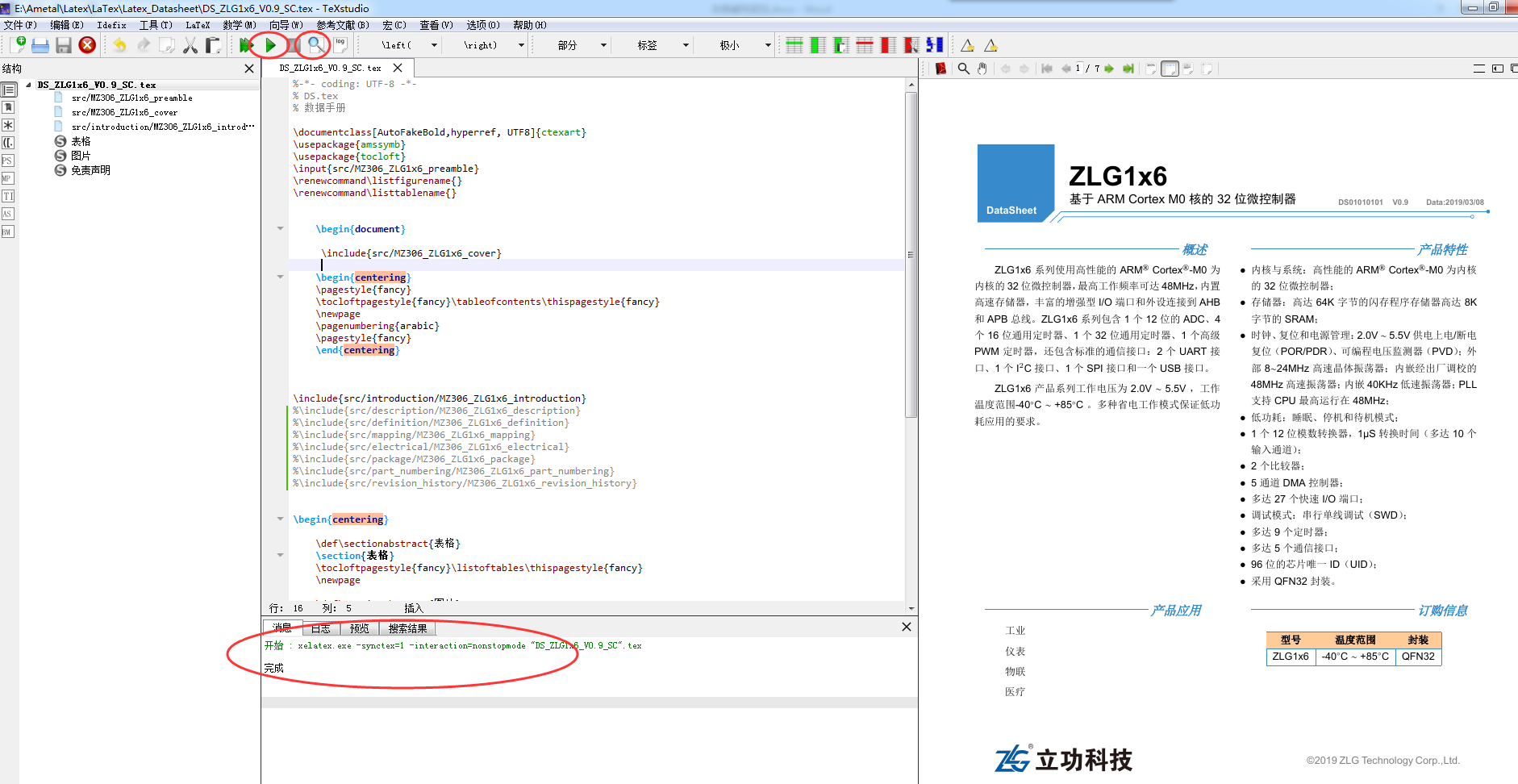


图2.7 编译工程生成pdf

### LaTeX优缺点

LaTeX总会拿来和一些“所见即所得”（What You See Is What You Get）的文字处理和排版工具比较优缺点。这种比较并不合适，每种工具都有值得使用的原因，这里总结了一些 LaTeX的优点：

* 专业的排版输出，产生的文档看上去就像“印刷品”一样；
* 方便而强大的数学公式排版能力；
* 绝大多数时候，只需专注于一些组织文档结构的基础命令，无需（或很少）操心文档的版面设计；
* 很容易生成复杂的专业排版元素，如脚注、交叉引用、参考文献、目录等；
* 强大的扩展性，世界各地的人开发了数以千计的 LaTeX宏包用于补充和扩展 LaTeX的功能；
* LaTeX促使用户写出结构良好的文档，而这也是 LaTeX存在的初衷；
* LaTeX依赖的 TEX 排版引擎和其它软件是跨平台、免费、开源的。无论用户使用的是 Windows， macOS（OS X）， GNU/Linux 还是 FreeBSD 等操作系统，都能轻松获得和使用这一强大的排版工具；

同时，LaTeX的缺点也是显而易见的：

* 入门门槛高；
* 排查错误困难，可以提示错误，但没有调试机制；
* 样式定制困难。 LaTeX提供了一个基本上良好的样式，为了让用户不去关注样式而专注于文档结构，但如果想要改进 LaTeX生成的文档样式则是十分困难；
* 相比“所见即所得”的模式有一些不便，为了查看生成的文档，用户总要不停地编译。

LaTeX的源代码本质上是文本文件，哪怕用 Windows 的记事本或者 Linux 的 gedit 等简单的编辑器，也可以编写一份 LaTeX源代码并编译出文档来。专用于编辑 LaTeX源代码的编辑器如 TeXworks / TeXstudio / WinEdt 等提供了一些语法高亮、命令补全等功能，以及调用排版引擎的一些按钮。除了文字本身， LaTeX源代码之中还包括大量的命令，用在排版公式、划分文档结构、控制样式等等不同的地方。

### LaTeX命令和环境

LaTeX命令以反斜线 \ 开头，为以下两种形式之一：

* 反斜线和后面的一串字母，如 \LaTeX。以任意非字母符号（空格、数字、标点等）作为分隔符；
* 反斜线和后面的一个非字母符号，如 \$。无需分隔符。

要注意 LaTeX命令是对大小写敏感的，比如输入 \LaTeX命令可以生成错落有致的 LaTeX字母组合，但输入 \LATeX或者 \LATEX什么都得不到，还会报错。

字母形式的 LaTeX命令忽略其后的所有空格。如果要人为引入空格，需要在命令后面加一对括号阻止其忽略空格。

大多数的 LaTeX命令是带一个或多个参数，每个参数用花括号 { 和 } 包裹。有些命令带一个或多个可选参数，以方括号 [ 和 ] 包裹。还有些命令在命令名称后可以带一个星号 \*，带星号和不带星号的命令效果有一定差异。

LaTeX还引入了环境的用法，用以令一些效果在局部生效，或是生成特定的文档元素。 LaTeX环境的用法为一对命令 \begin 和 \end，实例详见程序清单2.1。

程序清单2.1 局部生效源码示例

\begin{ environment name }

\ environment name {单次转换模式：A/D转换在指定通道完成一次转换}

\ environment name {单周期扫描模式：A/D转换在所有指定通道完成一个周期(从低序号通道到高序号通道)转换}

\ environment name {.....}

\ environment name {.....}

\end{ environment name }

其中 （environment name）为环境名， \begin 和 \end 中填写的环境名应当一致。 \begin 在（environment name） 后可以带一个或多个参数，甚至可选参数。环境允许嵌套使用。除了 LaTeX环境之外，花括号本身也起到分组的作用。

### LaTeX源码结构

LaTeX源代码以一个 \documentclass 命令作为开头，它规定了文档使用的文档类：

\documentclass{...}

LaTeX提供的文档类有 article, book, report等，在其基础上派生的一些文档类，如支持中文排版的 ctexart / ctexbook / ctexrep，或者有其它功能的一些文档类，如 moderncv / beamer 等，LaTeX提供的文档类型及具体说明详见表2.1。

表2.1 文档类型说明

|  |  |
| --- | --- |
| 文档类型 | 说明 |
| Article | 文章格式的文档类，广泛用于科技论文、报告、说明文档等； |
| Report | 长篇报告格式的文档类，具有章节结构，用于综述、长篇论文、简单的书籍等； |
| Book | 书籍文档类，包含章节结构和前言、正文、后记等结构； |
| Proc | 基于 article 文档类的一个简单的学术文档模板； |
| Slides | 幻灯格式的文档类，使用无衬线字体； |
| Minimal | 一个极精简的文档类，只设定纸张大小和基本字号，用作代码测试的最小工作示例。 |

紧接着可以用 \usepackage 命令调用宏包：

\usepackage{...}

在编写 LaTeX源代码时，常常会发现 LaTeX的基础功能不能满足所有需求，比如排版复杂的表格、插入图片、增加颜色甚至超链接等等。这时需要依赖一些扩展来增强或补充 LaTeX的功能，这些扩展称为宏包。\usepackage 的参数里可以使用不止一个宏包，多个宏包用逗号隔开，例如：

% 一次性载入三个排版表格常用的宏包

\usepackage{tabularx,makecell,multirow}

再接着，需要用以下一对命令来标记正文内容的开始位置和结束位置，而将正文内容写入其中：

\begin{document}

\end{document}

在 \documentclass 和 \begin{document} 之间的位置称为导言区，除了使用 \usepackage调用宏包之外，一些对文档的全局设置命令也在这里使用。当然也可以什么都不写，一个宏包都不调用，一切视自己需求。

### 文件类型说明

除了需要编写的源代码 .tex 文件，还可能接触到形形色色的文件。本节简单介绍一下这些文件。

每个宏包和文档类都是带特定扩展名的文件，除此之外也有一些文件出现于 LaTeX模板中，例如：

* .sty 宏包文件，宏包的名称就是去掉扩展名的文件名；
* .cls 文档类文件，同样地，文档类名称就是文件名；
* .bib BIBTEX 参考文献数据库文件；
* .bst BIBTEX 用到的参考文献格式模板。

LaTeX在编译过程中生成相当多的辅助文件和日志。一些功能如交叉引用、参考文献、目录、索引等，需要先编译生成辅助文件，然后再次编译时读入辅助文件得到正确的结果，所以复杂的LaTeX源代码可能要编译多次：

* .log 排版引擎生成的日志文件，供排查错误使用；
* .aux LaTeX生成的主辅助文件，记录交叉引用、目录、参考文献的引用等；
* .toc LaTeX生成的目录记录文件；
* .lof LaTeX生成的图片目录记录文件；
* .lot LaTeX生成的表格目录记录文件；
* .bbl BIBTEX 生成的参考文献记录文件；
* .blg BIBTEX 生成的日志文件；
* .idx LaTeX生成的供 makeindex 处理的索引记录文件；
* .ind makeindex 处理 .idx 生成的格式化索引记录文件；
* .ilg makeindex 生成的日志文件；
* .out hyperref 宏包生成的 PDF 书签记录文件。

### 文件组织方法

LaTeX支持将大规模的源码分成若干个文件，而不是写到一堆，尤其是针对开发芯片手册，对于芯片手册而言，动辄数千页，需要按章节，把它分成多个文件，每个文件可以单独编写和编译，确认各章内容准确无误，最后整合成一篇文档。

LaTeX提供了命令 \include 用来在源代码里插入文件，例如：

\include{<filename>}

<filename>为文件名，如果和要编译的主文件不在同一个路径中，需要加上相对或绝对路径，例如：

\include{src/abbreviation/MZ306\_ZLG1x6\_ABB} //相对路径

\include{D：/src/B&A/MZ306\_ZLG1x6\_B&A} //绝对路径

<filename> 可以不带扩展名，此时默认为 .tex；其它文件必须带扩展名。值得注意的是 \include 在读入 <filename> 之前会另起一页。有的时候并不需要这样，而是用 \input 命令，它纯粹是把文件里的内容插入：

\input{<filename>}

另外 LaTeX提供了一个 \includeonly 命令来组织文件，用于导言区，指定只载入某些文件，如下所示：

\includeonly{<filename1>,<filename2>,<filename3>....}

注：导言区使用了 \includeonly 后，正文中不在其列表范围的 \include 命令不会起效。

# 手册命名及版本管理规范

## 手册命名及版本管理规范

### 芯片手册类别

芯片相关手册目前主要有4类，数据手册、用户手册、勘误手册及应用笔记。

* 数据手册：数据手册主要包含芯片简介、特性介绍、器件选型、电气参数、机械尺寸等内容，打开工程编译即为数据手册模版，数据手册模版链接地址为： <http://gitlab/ametal/public_resource/tree/master/ZLGMCU_sheet/Latex_Datasheet>，模版样式首页详见图3.1。

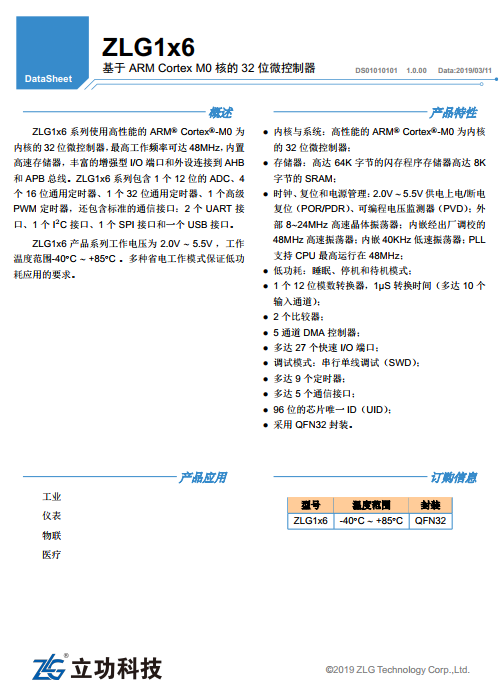


图3.1 芯片数据手册模版首页

* 用户手册：指导用户如何使用，涉及到一些比较深的内容，例如内部寄存器介绍，通信协议等，打开工程编译即可得到用户手册模版，用户手册模版链接地址为：<http://gitlab/ametal/public_resource/tree/master/ZLGMCU_sheet/Latex_UserManual>，可到此获取用户手册模版，用户手册模版详见图3.2。

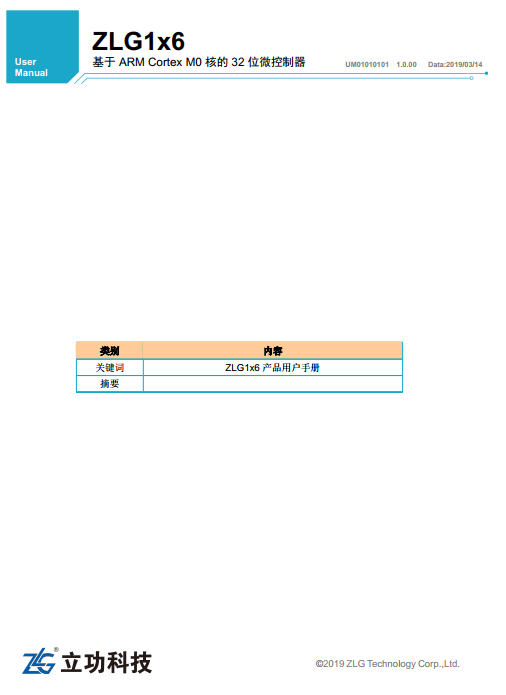


图3.2 芯片用户手册模版首页

* 勘误手册：指出芯片与当前手册描述不符或芯片本身存在的问题，并提供解决方法及建议，避免客户使用中出现问题，勘误手册模版获取地址为：<http://gitlab/ametal/public_resource/tree/master/ZLGMCU_sheet/Word_ErrataSheet> ，手册模版详见图3.3。

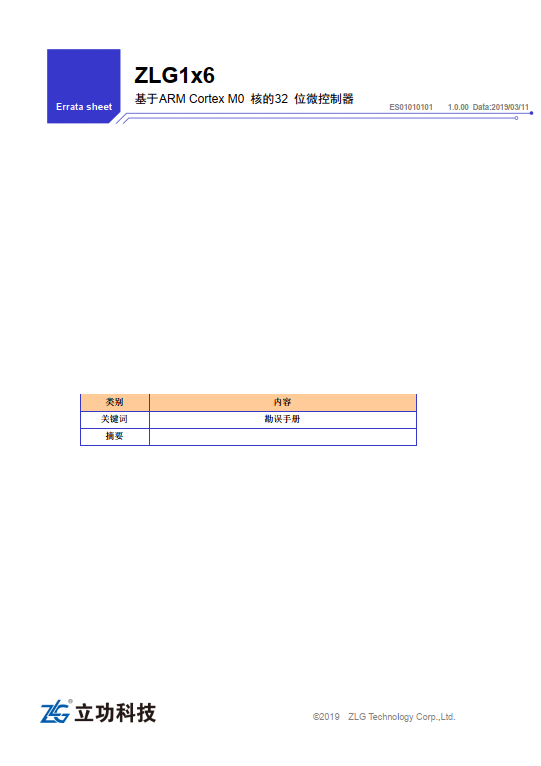


图3.3 勘误手册模版首页

* 应用笔记：提供应用开发阶段的软硬件设计参考，有针对性提供参考设计方法和示例，应用笔记模版获取地址为：<http://gitlab/ametal/public_resource/tree/master/ZLGMCU_sheet> ，应用笔记模版详见图3.4。

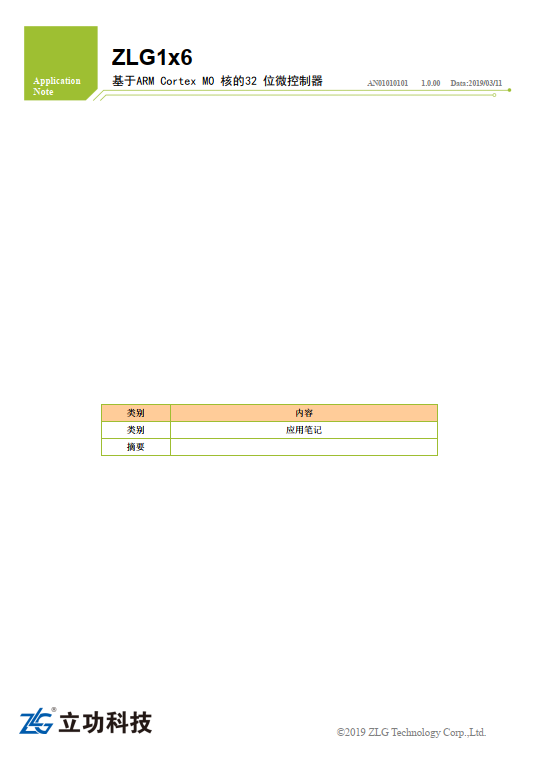


图3.4 应用笔记模版首页

### 芯片手册命名规范

手册采用统一的命名规范，格式如下：

【资料类型】 +自定义名称+版本号

例如，ZMF159xx芯片手册，数据手册可以命名为：【数据手册】ZMF159xx数据手册1.0.00 ；勘误手册可以命名为：【勘误手册】ZMF159xx勘误手册1.0.00；用户手册可命名为：【用户手册】ZMF159xx用户手册1.0.00 ，如所示图3.5。



图3.5 手册命名

### 版本管理及升序规则

在手册迭代过程中，引入版本控制来管理手册，这里定义了两项规则：

1. 版本起步控制规则

* 未发布的手册，版本编号从0.9.00开始，例如内部使用及编写中的手册，版本编号如下：【用户手册】ZMF159xx用户手册0.9.00，内部定版之后，后续的修改依据版本升序规则进行编号；
* 发布的手册，手册版本编号从1.0.00开始，发布版定版发布后，后续有修改，按照版本升序规则进行编码。

1. 版本升序规则

* 如果只是修改了部分问题，如优化段落，错别字等，不涉及目录更新（即标题、章节还是完全一样），这种修改，仅升级最后一位数字（修订号）；
* 增减小节（二级标题、三级标题…）升级次版本号，升级时，修订号归零；
* 增减章节，升级主版本号，升级时，次版本号及修订号归零；
* 当次版本号达到9时，再次升级时，升级主版本号，次版本号及修订号归零。

## 手册相关编码

### 手册编码规则

公司文档均统一编码，包括芯片手册，采用“文档类别+事业部+产品线+具体产品+文档序号”的编码方式，产品线编码由项目管理部门统一控制，具体产品由事业部经理发放，文档序号由项目负责人发放。

冗长的数字编码会造成口头校对的麻烦，特别是在编码没有隔断符号的时候。在创建本套编码规则的时候，已经考虑尽可能缩小编码的长度，但长达 8 位的数字还会造成口误。为了解决口头校对的问题，应当分段表达，以保证对方准确记录。读 UM01010215 时应当说UM， 0101， 0215。必须注意到以上规定仅限于口语表达，文书上依然不会采用这种方式。具体文档编码规则详见图3.6。

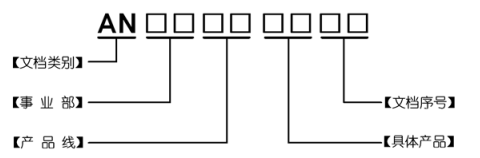


图3.6 编码规则

1. 文档类别

文档的类别定义、基本要素、特性及适用范围的详细说明见表3.1，基本要素仅供参考，建议文档撰写人仅作章节的删减，而不要变更章节名称，以保证公司文档结构整齐统一。

表3.1 文档类别

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别缩写 | 定义 | 基本要素 | 特性 | 适应范围 |
| DS | DS 是产品数据手册 Data Sheet 的缩写 | 包含目录、概述、特性、应用范围、选型信息、典型应用、电气参数、机械参数等； | 网上提供免费下载 | 提供足够的参数和详细选型依据； |
| UM | UM 是产品用户手册 User Manual的缩写 | 包含目录、概述、特性、应用范围、寄存器描述、通信协议等； | 网上提供免费下载 | 指导用户深入了解芯片功能，使用和开发产品； |
| ES | ES是产品勘误手册Errata Sheet的缩写 | 包含目录、概述、勘误信息、解决方法等； | 网上提供免费下载 | 指导客户开发过程中避开已知问题； |
| AN | AN是应用笔记Application Note的缩写 | 包含目录、概述、软硬件参考设计方法等； | 网上提供免费下载 | 提供客户开发环节中的参考设计； |

1. 事业部

编码范围是： 01～99。仅项目管理部门有权限修改与发布。01为周立功芯片相关编码，其他待定义。

1. 产品线

编码范围是： 01～99。仅事业部经理有权限修改与发布，同时必须知会项目管理部门。虽然“产品线+具体产品+文档序号”的编码方式有 9999 个产品容量，但实际上使用达不到这个理论值，所以事业部经理必须细分产品线。

对于自主芯片一块，默认使用产品线默认定义为01。

1. 具体产品

编码范围是： 01～99。仅项目负责人有权限修改与发布，同时必须知会项目管理部门。类似于产品线编码，对于型号非常接近的，或者衍生产品比较丰富的型号，一般视为同一个产品，避免具体产品号段被快速消耗。

对于自主芯片一块，目前具体产品编号主要如表3.2所示。

表3.2 自主芯片产品类别

|  |  |  |
| --- | --- | --- |
| 具体产品 | 产品类别 | 产品归类 |
| 01 | MCU | Coretex M0：ZLG116N32A、ZLG126N32A、ZML166N32A等  Coretex M3：ZLG217P64A、ZLG227P64A、ZMF159N48A等 |
| 02 | LDO | ZL6201、ZL6205、ZL6210、ZL6300、ZL6303等 |
| 03 | 逻辑驱动 | ZLG72128、ZLG7289、ZLG7290等 |
| 04 | 传感器 | RS01等 |
| 05 | 无线 | ZSN600等 |

1. 文档序号

编码范围是： 01～99，项目负责人内部掌握。

### 手册编码示例

手册编码在芯片手册的首页填入，以ZLG116芯片的数据手册为例，属于周立功芯片相关，编号01；产品线为自有芯片，编号01；具体产品属于MCU一类，编号为01；文档序号，该手册为第一篇，序号注册为01，则该手册的编码为DS01010101，如图3.7所示。



图3.7 手册编码示意图

### 勘误信息编号规则

勘误手册中，需要给每一项勘误信息编号，方便管理。命名规则如图3.8所示。

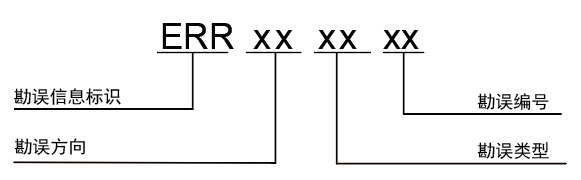


图3.8 勘误信息编码

1. 勘误信息标识：ERR，作为勘误信息标识，放在编码起始位置；
2. 勘误方向：表明勘误的是何手册或者手册之外的勘误，如下所示：

* 01：对数据手册内容进行勘误；
* 02：对用户手册进行勘误；
* 03：芯片手册描述之外的勘误。

1. 勘误类型：具体勘误的类型，分类如表3.3所示。

表3.3 勘误类型编码表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 说明 | 编号 | 说明 |
| 01 | 电气特性相关 | 02 | 时钟特性相关 |
| 03 | 电源控制相关 | 04 | Flash操作相关 |
| 05 | 复位时钟控制相关 | 06 | GPIO相关 |
| 07 | ADC相关 | 08 | COMP相关 |
| 09 | DMA相关 | 10 | RTC相关 |
| 11 | TIM相关 | 12 | UART相关 |
| 13 | 看门狗相关 | 14 | I2C相关 |
| 15 | SPI相关 | 16 | USB相关 |
| 17 | CAN相关 | 18 | BKP相关 |
| 19 | 以太网相关 | 20 | SDIO相关 |
| 21 | CRC相关 | 22 | （其他往后添加） |

1. 勘误编号：

勘误编码，勘误编码默认从01开始编码，勘误编码前的勘误类型及勘误方向一致时，新添加的勘误开始累加勘误编码。例如：对ZLG217P64A的用户手册关于ADC的两条勘误信息，第一条可以编码为ERR020301，第二条可以编码为ERR020302，以此类推。

# 手册编写规范

## 手册LaTeX工程结构

### 文件结构说明

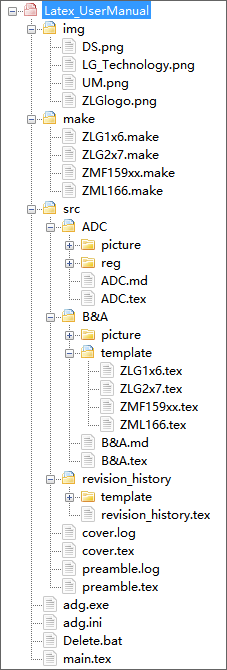


图4.1 用户手册工程文件结构

如图4.1所示，用户手册的工程文件包含三个文件夹：

1. img：包含了手册中固定的图片，如首页及页脚中所用到的立功科技的Logo；
2. make：包含了各个芯片的信息，支持的外设，及通道数量等，有关make文件的详细说明，请参照4.1.2小节；
3. src：此文件夹内基本包含了手册所有内容，首页的内容保存在源文件“cover.tex”中；页眉页脚、文字大小及格式都保存在源文件“preamble.tex”中；手册中的其他部分放在对应的文件夹中，一般每一个文件夹对应一个章节，编译出的手册中包含的章节由make文件决定；在这些文件夹内，又包含有三种文件夹：picture、reg以及template：

* picture包含了该章节会用到的图片；
* reg中为该章节中所有寄存器的描述的图片；
* 有些内容，如修订记录，全部放在一个源文件中会更加混乱，所以这部分会单独存放在对应tex文件，并放在template文件夹中。

在同级目录下，还有着四个文件，其中最主要的为“adg.exe”，此应用程序加载make文件后，会将src内对应的源文件中的相关部分提取出来，保存至另一个tex文件，用来生成make文件对应芯片的手册，关于“adg.exe”的使用方法，详见4.1.3小节；

其他三个文件的说明如下：

* “adg.ini”为“adg.exe”的使用信息；
* “Delete.bat”用于删除生成手册过程中所产生的临时文件；
* “main.tex”为手册的主框架，里面决定了首页、目录、章节内容的顺序。

### Make文件说明

Make文件中，包含两个部分：info和file。

1. info

在info中以一种给变量赋值的方式，保存了该芯片所有的信息，包括名字、主频、flash大小、定时器个数等，在用户手册的Make文件中，还添加了各个外设寄存器的起始地址；

这些信息的用法有两个：

* 用于判断，达到条件编译的效果；格式为%` if … `%，可以使用的符号有括号、与、或，在结束的地方加上%` `%，实现end if的效果，详见程序清单4.1，需要注意的是它不支持if嵌套！

程序清单4.1 条件编译示例

1 %` if !partCode = MZ308 `%

2 \item{最高 12 位可编程分辨率的 SAR ADC，多达\#{adcChal}\#路外部输入通道。}

3

4 %` if partCode = MZ308 `%

5 \item{最高12位可编程分辨率的SAR ADC，其中ADC1多达12路外部输入通道和2路内部通道，ADC2多达12路外部输入通道。两个ADC共享8个外部通道，并各有4个独立的外部通道。}

6

7 %` if adcVer = 3 `%

8 \item{ADC1的内部信号源接内部温度传感器，ADC2的内部信号源接内部1.2V参考电压}

9

10 %` `%

* 直接替换，特定格式的变量名能在使用过程中替换成它们所对应的值；以wordName芯片名字为例，有两种方式能将info中wordName的值显示出来：

%` =wordName `%

\#{wordName}\#

推荐使用第二种，因为第一种能将变量值显示出来的同时，还有上述所说的end if的效果，可能会导致显示的内容错乱。

1. file

在file部分中，包含了生成该手册所用到的源文件，前三个已经固定，按照顺序为“main.tex”、“cover.tex”以及“preamble.tex”，分别为手册主框架、手册首页和手册页面模板文件，后面的部分为该手册所包含的章节，一般来说，每一个章节对应一个文件，手册中增减章节、调整章节顺序都应该在file部分完成。

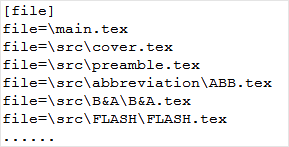


图4.2 make文件file部分

### 芯片手册工程生成

双击打开“sdg.exe”，点击装入Make文件，选择ZLG1x6的make文件，如图4.3所示，左边的部分显示了ZLG1x6.make中的info部分的内容，右边显示了file部分的内容，点击运行，提示：“脚本运行成功，目标文件已产生”后，在当前路径能找到一个新的tex文件：UM\_ZLG1x6\_V0.90\_SC.tex。

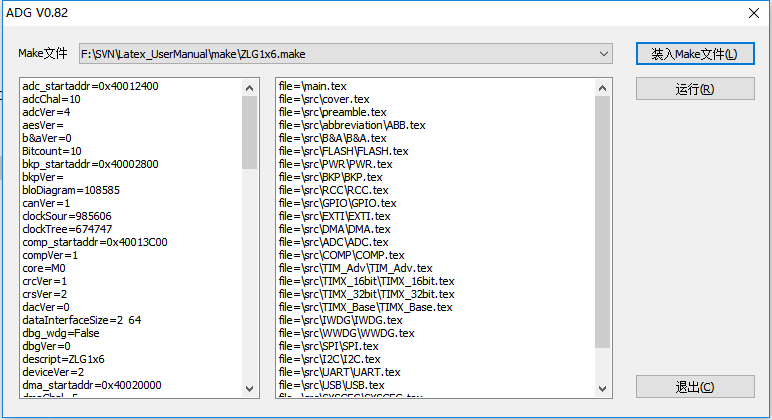


图4.3 adg.exe运行界面

UM\_ZLG1x6\_V0.90\_SC.tex是由main.tex编辑后重命名得来的，该文件的内容详见程序清单4.2，该文件内容作用如下：

* 第4行为该手册页面模板文件；
* 第10行则为该手册的首页；
* 第11行为该手册的修订记录；
* 12~18行为手册的目录；
* 20~43行会加入make文件中file部分中preamble文件后的内容；
* 45~57行为手册的图、表的目录；
* 59行后为免责声明。

程序清单4.2 main.tex文件内容

1 \documentclass[AutoFakeBold,hyperref, UTF8]{ctexart}

2 \usepackage{amssymb}

3 \usepackage{tocloft}

4 \input{src/MZ306\_ZLG1x6\_preamble}

5 \renewcommand\listfigurename{}

6 \renewcommand\listtablename{}

7

8 \begin{document}

9

10 \include{src/MZ306\_ZLG1x6\_cover}

11 \include{src/revision\_history/MZ306\_ZLG1x6\_revision\_history}

12 \begin{centering}

13 \pagestyle{fancy}

14 \tocloftpagestyle{fancy}\tableofcontents\thispagestyle{fancy}

15 \newpage

16 \pagenumbering{arabic}

17 \pagestyle{fancy}

18 \end{centering}

19

20 \include{src/abbreviation/MZ306\_ZLG1x6\_ABB}

.. ......

43 \include{src/part\_numbering/MZ306\_ZLG1x6\_part\_numbering}

44

45 \begin{centering}

46

47 \def\sectionabstract{表格}

48 \section{表格}

49 \tocloftpagestyle{fancy}\listoftables\thispagestyle{fancy}

50 \newpage

51

52 \def\sectionabstract{图片}

53 \section{图片}

54 \tocloftpagestyle{fancy}\listoffigures\thispagestyle{fancy}

55 \newpage

56

57 \end{centering}

58

59 \def\sectionabstract{免责声明}

60 \section{免责声明}

.. ......

73

74 \end{document}

用TeXstudio打开此文件，点击构建并查看按钮（快捷键F5），就能生成一篇完整的手册。

### 章节源码及中间源码说明

无论是用户手册还是数据手册，章节源码都在src文件夹内，每当用“sdg.exe”装入make文件生成目标文件时，都会在章节源码所在的文件夹内生成编译对应手册所需的中间源码。详见图4.4，“definition.tex”为章节源码，“MT304\_ZLG2x7\_definition.tex”为ZLG2x7数据手册所需的中间源码。

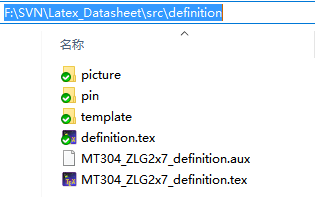


图4.4 数据手册的definition文件夹

中间源码是直接从章节源码抽取出来的，是实际参与到工程中的文件，修改中间源码可以直接作用于最终PDF生成。但是不建议在中间源码进行修改，或者说禁止在中间源码进行修改，应该到源码处进行修改，修改之后，再使用adg.exe生成新的中间源码，这是因为最后不会保留中间源码，只保留一份最终章节源码，用来抽取，每次手册修改生成之后，删除中间源码。下一次生成工程时，重新抽取生成中间源码来编译。

因为源码涉及到多颗芯片，所以修改某一颗芯片的相关内容时，注意不要对其他的芯片的内容产生影响。

## 手册编写规范

### 模板规范

如图4.5所示，每个手册的首页上面应该说明这是什么芯片的手册，以及版本号，生成日期。

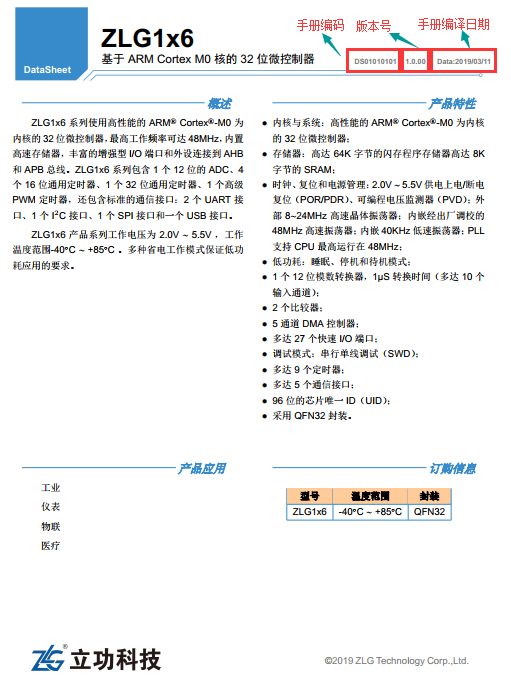


图4.5 手册模版首页说明

### 图片引用规范

普通图片的引用有两种命令，分别对应宽的图片和窄的图片；插入图片的命令为：

* \fullPic{图片题注}{图片路径}{图片标签}：插入宽图片；
* \smallPic{图片题注}{图片路径}{图片标签}：插入窄图片；

宽图片的宽度为490pt，窄图片的宽度为392pt，图片中的中文字体为宋体，英文字体为Arial或Times New Roman，字体大小自行调整。应确保图片风格基本一致。插入普通图片的示例见图4.6。

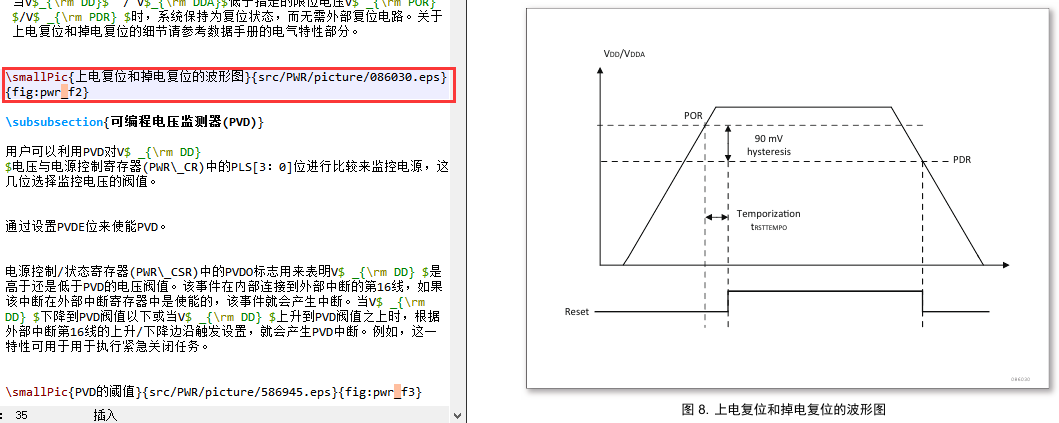


图4.6 插入普通图片示例

插入寄存器的说明图片时，使用固定的命令：

\regPic{图片路径}

寄存器的说明图片宽度固定位490pt，字体使用Arial，建议在增添或修改这些图片时，复制一份，直接在这一份的基础上修改，确保字体大小一致。插入寄存器说明图片的示例见图4.7

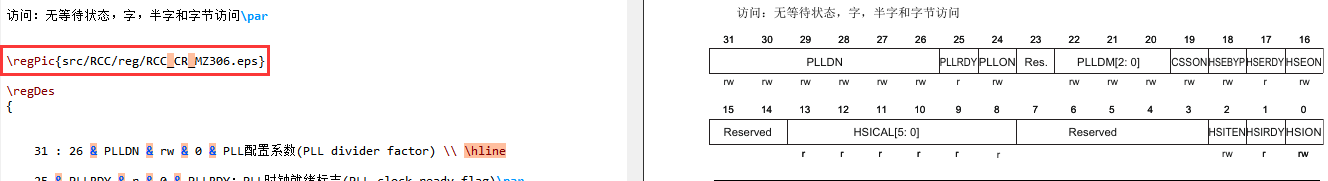


图4.7 插入寄存器说明图片

### 图片编辑规范

手册中的插图建议使用后缀为.eps格式的位图，可采用AI或Photoshop等图片编辑工具，方便编辑和引用，如图4.8所示。对图片的尺寸没有具体的要求，看实际图片需求而定，图片编辑部分要求如下：

* 字体：黑体（中）、Arial（西）；
* 字号： 一般文本为9.46pt，特殊文本看具体情况微调。

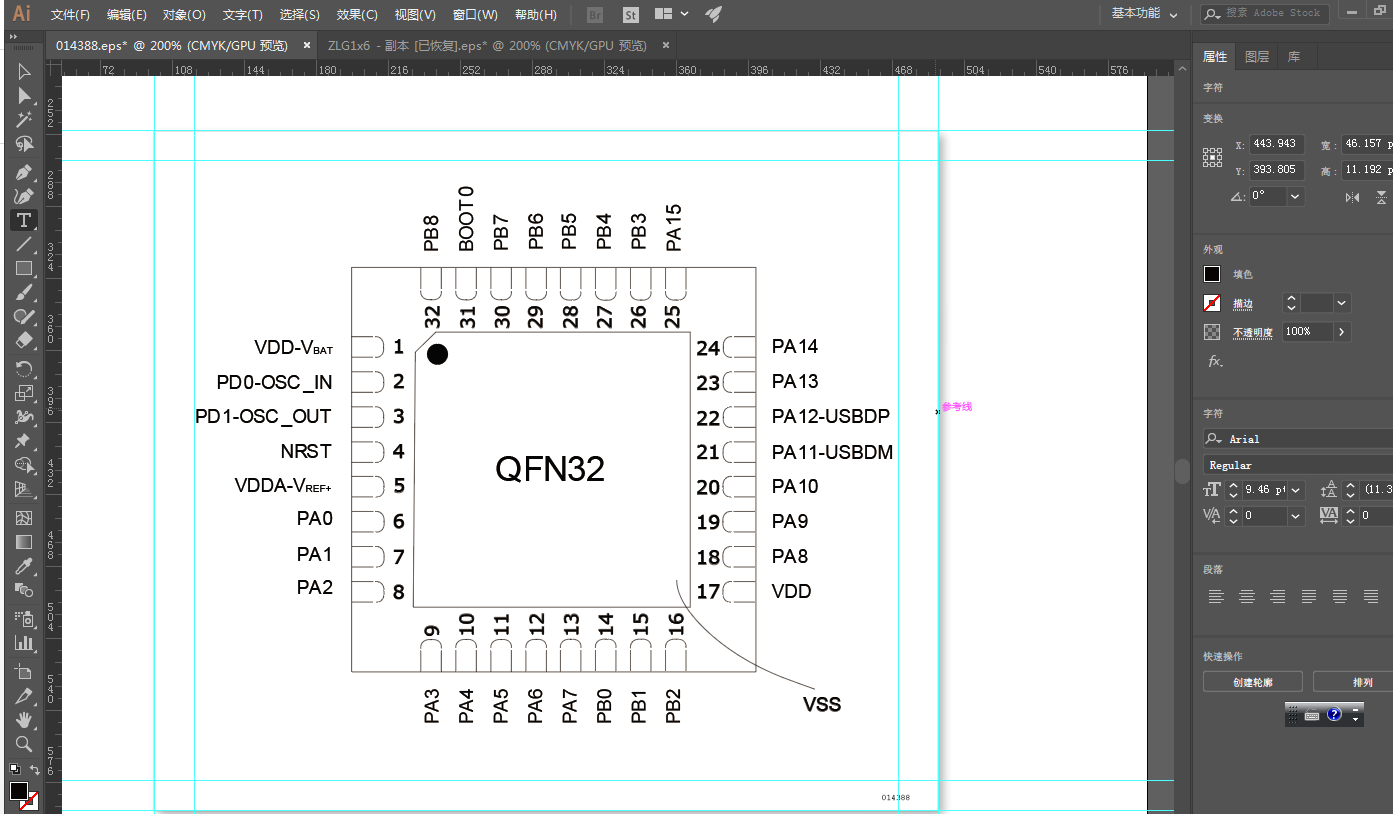


图4.8 编辑图片

### 表格编写及规范

表格可分为三种：普通表格、寄存器概览表格以及寄存器位描述表格；寄存器概览表格和寄存器位描述表格的格式都已在模板中写好，使用时调用对应的命令即可。

1. 普通表格

普通表格的表头必须加上底纹，表头文字加粗；如程序清单4.3所示，生成的表格详见图4.9，每个表格必须加上标签便于引用，如“\label{tab:T10777}”。

程序清单4.3 示例表格

1 \begin{center}

2 \tabpro{0}

3 \begin{longtable}{|m{2cm}<{\centering}|m{1.6cm}<{\centering}|

4 m{1.4cm}<{\centering}|m{1.4cm}<{\centering}|}

5 \caption{示例表格}\label{tab:T10777}

6 \\ \hline\hline

7 \rowcolor{mygray}

8 \bfseries 示例1 & \bfseries 示例2 & \bfseries 示例3 & \bfseries 示例4 \\ \hline\hline

9 \endfirsthead

10 \hline\hline

11 \rowcolor{mygray}

12 \bfseries 示例1 & \bfseries 示例2 & \bfseries 示例3 & \bfseries 示例4 \\ \hline\hline

13 \endhead

14 A & B & C & D \\ \hline

15 E & F & G & H \\ \hline

16 \end{longtable}

17 \end{center}



图4.9 生成的表格

1. 寄存器概览表格

寄存器概览表格一共有5列，分别为：偏移地址、名称缩写、寄存器名称、复位值以及该寄存器对应章节；插入寄存器概述的示例详见图4.10。

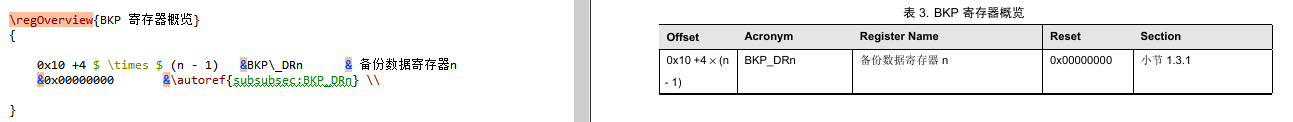


图4.10 寄存器概览

1. 寄存器位描述表格

寄存器位描述表格共有5列，分别为：寄存器位、该位名称、读写类型、复位值以及具体描述，插入寄存器位描述表格的示例详见图4.11。

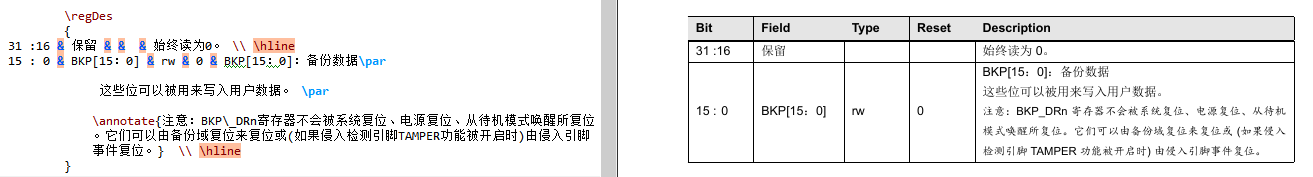


图4.11 寄存器描述

### 交叉引用方法及规范

在需交叉引用的对象已使用“\lable{标签名}”创建标签时，直接在需要引用的地方输入“\autoref{标签名}”即可完成引用，详见图4.12。

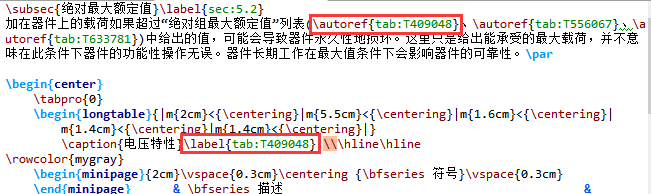


图4.12 交叉引用

### 公式编写及规范

插入公式时要求居中，并使用默认字体，插入公式的示例详见图4.13，公式中使用的命令都能在菜单栏中的“数学”栏中找到，详见图4.14。

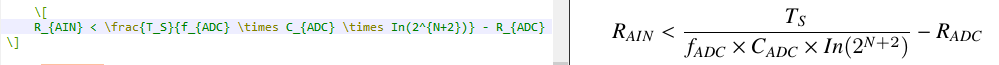


图4.13 插入公式示例

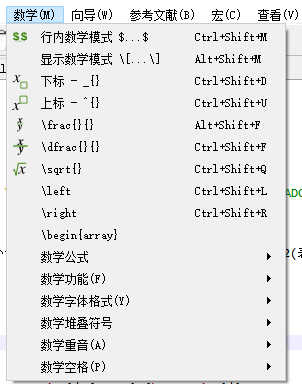


图4.14 数学栏

### 历史修订记录填写规范

修订记录在“\src\revision\_history\template”路径下对应的以芯片命名的源文件里修改，每次对手册进行修改，都必须添加修订记录，修订记录中要准确描述此次修改的内容，修改的时间以及修改人，将这部分信息添加到“\iffalse”后，再在上面的表格中添加版本更新记录，具体格式请参照图4.15。版本号的命名规则请参照3.2小节。

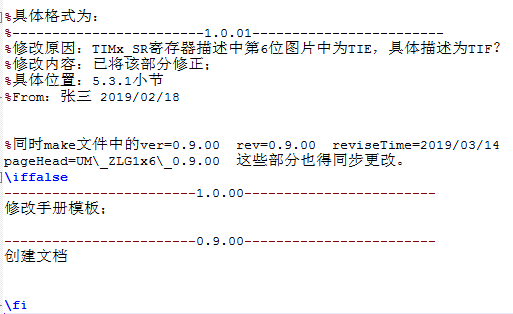


图4.15 修订记录格式

## LaTeX常用语法

### 空格与分段

文档编写中，经常需要实现空格和分段，在LaTeX源码中，空格键和Tab键输入空白字符视为“空格”，连续若干个空白字符视为一个空格，开头的空格忽略。

行末的回车视为一个空格，连续两个回车视为空行将文字分段，多个空行视为一个空行，也可以在末尾使用分段命令。

\par

### 注释

LaTeX用 % 字符作为注释。在这个字符之后直到行末，所有的字符都被忽略，行末的回车也不引入空格，示例详见图4.16。

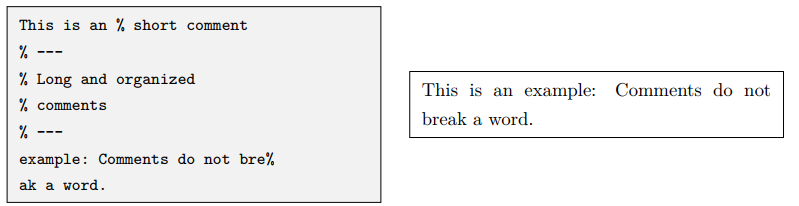


图4.16 注释示例

### 特殊字符表示

以下字符在 LaTeX里有特殊用途，如 % 表示注释， $、 ^、 \_ 等用于排版数学公式， & 用于排版表格等等。直接输入这些字符得不到对应的符号，还往往会出错：# $ % & { } \_ ^ ~ \。如果想要输入以上符号，需要使用以下带反斜线的形式输入，详见示例图4.17。



图4.17 特殊字符使用示例

### 标点符号使用

中文的标点符号（都是全角的非 ASCII 字符）一般来讲不用特别操心，只要使用中文输入法输入即可。而在英文标点的输入上，有许多需要留意的地方。

1. 引号

LaTeX的单引号 ‘ ’ 用 ‘ 和 ’ 输入；双引号 “ ” 用 ‘‘ 和 ’’ 输入（虽然 " 也能表示双引号，但没有合适的单个符号用来表示前双引号，所以不常用在正文），示例详见图4.18。



图4.18 引号使用示例

1. 连字号和破折号

LaTeX中有三种长度的“横线”可用：连字号、短破折号（en-dash）和长破折号（em-dash）。它们分别有不同的用途：连字号 - 用来组成复合词；短破折号 – 将数字连接表示范围；长破折号— 作为破折号使用，示例详见图4.19。

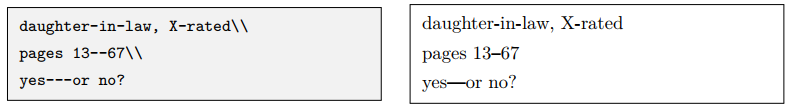


图4.19 连字号及破折号使用示例

1. 省略号

LaTeX提供了命令 \ldots 来生成省略号，相对于直接输入三个点的方式更为合理。 \ldots和 \dots 是两个等效的命令，示例详见图4.20



图4.20 省略号使用示例

1. 下划线

LaTeX定义了\underlined 来实现为文字添加下划线，示例详见图4.21。



图4.21 添加下划线示例

### 断行及断页

ATEX 将文字段落在合适的位置进行断行，尽可能做到每行的宽度一致，并且单词间距匀称。文字段落和公式、图表等内容从上到下顺序排布，并在合适的位置断页，分割成匀称的页面。

在绝大多数时候，无需自己操心断行和断页。但偶尔会遇到需要手工调整的地方，如跨页的表格，部分内容可能出现上下两页风格，不美观，这时候通常需要手动调整，以及长表格内容太长，使得表格不协调，或单词、链接等过长，没有自动断行，这时候需要手动断行。

如果需要断行和断页，可以使用如下命令：

\\ or \newline 断行

\newpage or \clearpage 断页

\\ 也在表格、公式等地方用于分行，而 \newline 只用于文本段落中。

有时候不满足于 LaTeX默认的断行和断页位置，需要进行微调，就要用以下命令告诉LaTeX哪些地方适合断页，哪些地方不适合，命令如下所示：

\linebreak [< >] \nolinebreak [< >]

\pagebreak [< >] \ nopagebreak [< >]

以上命令都带一个可选参数，用数字 ⟨n⟩ 代表适合/不适合的程度，取值范围为 0-4，不带可选参数时，缺省为 4。比如 \linebreak[3] 意味着此处在断行时优先考虑； \nopagebreak 或\nopagebreak[4] 意味着禁止在此处断页。

以上命令适合给出优先考虑断行断页/禁止断行断页的位置，但不适合直接拿来断行或断页，使用 \newline 或 \newpage 等是更好的选择。

### 章节和目录

一篇结构化的、条理清晰文档一定是层次分明的，通过不同的命令分割为章、节、小节。 LaTeX的三个标准文档类 article、 report 和 book1提供了一系列命令，用以划分章节、生成章节标题并自动编号，命令如下所示：

\chapter{<title>} \section{<title>}

\subsection{<title>} \subsubsection{<title>}

\paragraph{<title>} \subparagraph{<title>}

其中 \chapter 只在 book 和 report 文档类有定义。\part 命令用以将整个文档分割为大的分块，但不影响 \chapter 或 \section 等的编号：\part{⟨title⟩}上述命令除了生成带编号的标题之外，还向目录中添加条目，并影响页眉页脚的内容。每个命令有两种变体：

* 带可选参数的变体： \section[⟨short title⟩]{⟨title⟩}标题使用 ⟨title⟩ 参数，在目录和页眉页脚中使用 ⟨short title⟩ 参数；
* 带星号的变体： \section\*{⟨title⟩}标题不带编号，也不成目录项和页眉页脚。较低层次如 \paragraph 和 \subparagraph 即使不用带星号的变体，生成的标题默认也不带编号，事实上，除 \part 外：
* article 文档类带编号的层级为 \section / \subsection / \subsubsection 三级；
* report/book 文档类带编号的层级为 \chapter / \section / \subsection 三级。

### 目录生成

在 LaTeX中生成目录非常容易，只需在合适的地方使用命令：

\tableofcontents

这个命令会生成单独的一个章节。正确生成目录项，一般需要编译两次源代码。有时用 \chapter\* 或 \section\* 这样不生成目录项的章节标题命令，而又想手动生成该章节的目录项，可以在标题命令后面使用：

\addcontentsline{toc}{<level>}{<title>}

其中 <level> 为章节层次 chapter 或 section 等， <title> 为出现于目录项的章节标题。

### 文档结构划分

所有标准文档类都提供了一个 \appendix 命令将正文和附录分开，使用 \appendix 后，最高一级章节改为使用拉丁字母编号，从 A 开始。

book 文档类还提供了前言、正文、后记结构的划分命令：

* \frontmatter 前言部分，页码为小写罗马字母格式；其后的 \chapter 不编号；
* \mainmatter 正文部分，页码为阿拉伯数字格式，从 1 开始计数；其后的章节编号正常；
* \backmatter 后记部分，页码格式不变，继续正常计数；其后的 \chapter 不编号。

### 文本对齐

center、 flushleft 和 flushright 环境分别用于生成居中、左对齐和右对齐的文本环境。

\begin{center} . . . \end{center} 居中

\begin{flushleft} . . . \end{flushleft} 左对齐

\begin{flushright} . . . \end{flushright} 右对齐

除此之外，还可以用以下命令直接改变文字的对齐方式，如图4.22所示。

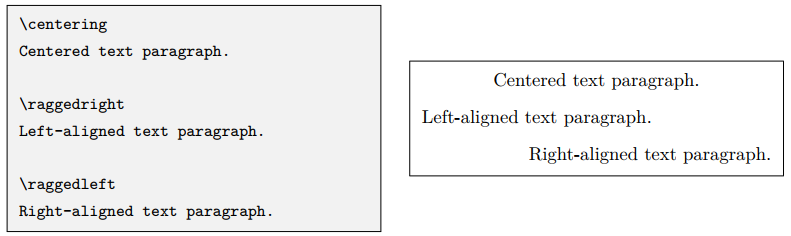


图4.22 文本对齐

三个命令和对应的环境经常被误用，有直接用所谓 \flushleft 命令或者 raggedright 环境的，都是不甚严格的用法（即使它们可能有效）。有一点可以将两者区分开来： center 等环境会在上下文产生一个额外间距，而 \centering 等命令不产生，只是改变对齐方式。比如在浮动体环境 table 或 figure 内实现居中对齐，用 \centering 命令即可，没必要再用 center 环境。

### 交叉引用

交叉引用是 LaTeX强大的自动排版功能的体现之一。在能够被交叉引用的地方，如章节、公式、图表、定理等位置使用 \label 命令：

\label{<label-name>}

之后可以在别处使用 \ref 或 \pageref 命令，分别生成交叉引用的编号和页码，详见图4.23。

\ref{<label-name>} \pageref{<label-name>}

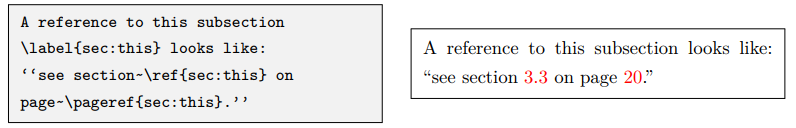


图4.23 交叉引用示例

### 字体样式命令

LaTeX提供了两组修改字体的命令，详见图4.24。其中诸如 \bfseries 形式的命令将会影响之后所有的字符，如果想要让它在局部生效，需要用花括号分组，也就是写成 {\bfseries ⟨sometext⟩} 这样的形式；对应的 \textbf 形式带一个参数，只改变参数内部的字体，更为常用。



图4.24 字体命令

### 字号命令

字号命令实际大小依赖于所使用的文档类及其选项。图4.25列出了这些命令在标准文档类中的绝对大小，单位为 pt。

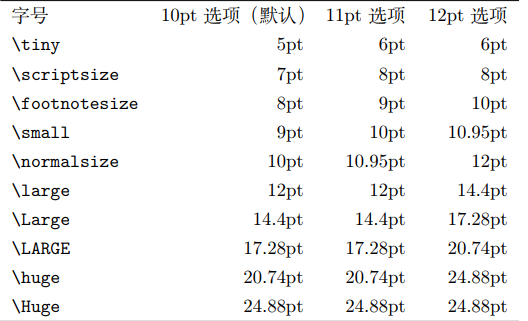


图4.25 字号大小

使用字号命令的时候，通常也需要用花括号进行分组，如同 \rmfamily 那样，使用示例详见图4.26。



图4.26 字号应用示例

LaTeX还提供了一个基础的命令 \fontsize 用于设定任意大小的字号，如下所示：

\fontsize{⟨size⟩}{⟨base line-skip⟩}

\fontsize 用到两个参数， ⟨size⟩ 为字号， ⟨base line-skip⟩ 为基础行距。图4.25中的命令也都各自设定了与字号对应的基础行距，大小为字号的 1.2 倍。如果不是在导言区， \fontsize 的设定需要 \selectfont 命令才能立即生效。L

LaTeX 排版用到的一些老式字体宏包适用图4.25给出的固定字号1，在使用任意的字号大小时往往会报一些警告。如果可能的话，应当先尽量使用命令设置字号。

### 行距设置

前文中提到过 \fontsize 命令可以为字号设定对应的行距，但很少那么用。更常用的办法是在导言区使用 \linespread 命令，如下所示：

\linespread{⟨factor⟩}

这里的 ⟨factor⟩ 是在基础行距上而不是字号上乘以一个因子。大部分时候，默认的基础行距是 1.2 倍字号大小（参考 \fontsize 命令），因此设置 1.5 倍行距的命令 \linespread{1.5} 意味着最终行距为 1.8 倍的字号大小。

如果不是在导言区全局修改，而想要局部地改变某个段落的行距，需要用 \selectfont 命令使 \linespread 命令的改动立即生效，如图4.27所示。

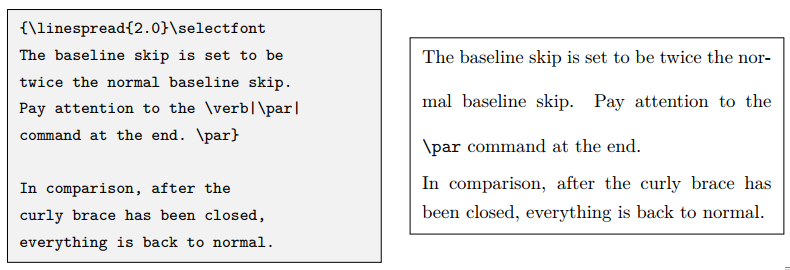


图4.27 行距设置

字号的改变是即时生效的，而行距的改变直到文字分段时才生效。如果想改变某一部分文字的行距，那么不能简单地将文字包含在花括号内，如图4.28所示。

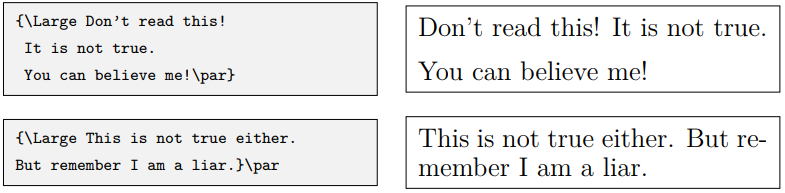


图4.28 局部行距

### 段落缩进格式

以下长度分别为段落的左缩进、右缩进和首行缩进：

\setlength{\leftskip}{20pt}

\setlength{\rightskip}{20pt}

\setlength{\parindent}{2em}

它们和设置行距的命令一样，在分段时生效。LaTeX默认在段落开始时缩进，长度为上述命令设置的 \parindent。如果你在某一段不想使用缩进，可使用某一段开头使用 \noindent 命令。相反地， \indent 命令强制开启一段首行缩进的段落。多个 \indent 命令可以累加缩进量。

LaTeX还默认在 \chapter、 \section 等章节标题命令之后的第一段不缩进4。如果不习惯这种设定，可以调用 indentfirst 宏包，令第一段的首行缩进照常。

### 垂直间距设置

在页面中，段落、章节标题、行间公式、列表、浮动体等元素之间的间距是 LaTeX预设的。比如 \parskip ，默认设置为 0pt plus 1pt。如果想要人为地增加段落之间的垂直间距，可以在两个段落之间的位置使用如下命令：

\vspace{⟨length⟩}

\vspace 的间距在一页的顶端或底端可能被“吞掉”，类似 \hspace 在一行的开头和末尾那样。对应地， \vspace\* 命令产生不会因断页而消失的垂直间距。 \vspace 也可用 \stretch 设置无限延伸的垂直长度。

在段落内的两行之间增加垂直间距，一般通过给断行命令 \\ 加可选参数，如 \\[6pt] 或\\\*[6pt]。 \vspace 也可以在段落内使用，如图4.29所示。

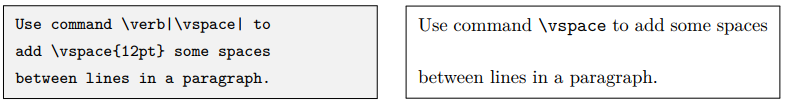


图4.29 垂直间距修改示例

另外 LaTeX还提供了\bigskip, \medskip, \smallskip 来增加预定义长度的垂直间距，如图4.30所示。

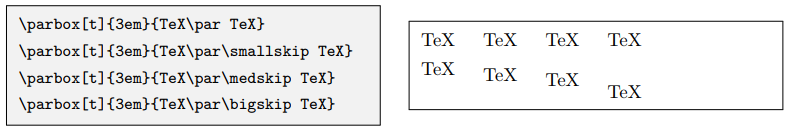


图4.30 垂直间距

### 水平间距

LaTeX默认为将单词之间的“空格”转化为水平间距。如果需要在文中手动插入额外的水平间距，详见图4.31，可使用如下命令：

\hspace{⟨length⟩}

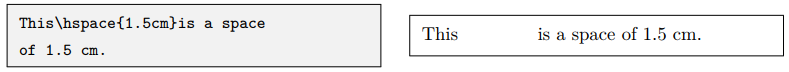


图4.31 水平间距示例

\hspace 命令生成的水平间距如果位于一行的开头或末尾，则有可能因为断行而被“吞掉”。可使用 \hspace\* 命令代替 \hspace 命令得到不会因断行而消失的水平间距。命令 \stretch{⟨n⟩} 生成一个特殊弹性长度，参数 ⟨n⟩ 为权重。它的基础长度为 0pt，但可以无限延伸，直到占满可用的空间。如果同一行内出现多个 \stretch{⟨n⟩}，这一行的所有可用空间将按每个 \stretch 命令给定的权重 ⟨n⟩ 进行分配，详见图4.32。

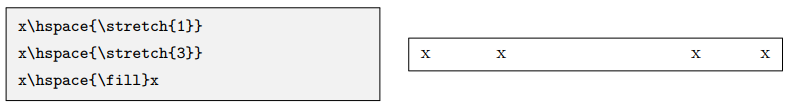


图4.32 水平间距示例

在正文中用 \hspace 命令生成水平间距时，往往使用 em 作为单位，生成的间距随字号大小而变。在数学公式中的 \quad 和 \qquad 命令，它们也可以用于文本中，分别相当于\hspace{1em} 和 \hspace{2em}，详见图4.33。

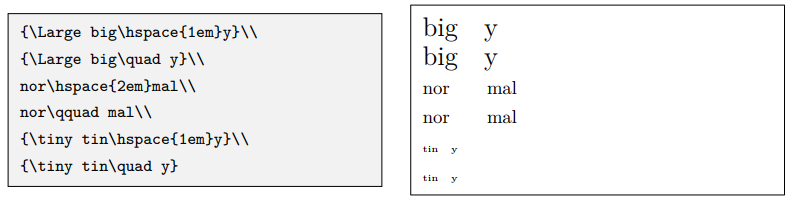


图4.33 水平间距

### 页边距设置

比如，符合 Microsoft Word 习惯的页面设定是 A4 纸张，上下边距 1 英寸，左右边距 1.25英寸，于是可以通过如下两种等效的方式之一设定页边距：

\usepackage[left=1.25in,right=1.25in,%

top=1in,bottom=1in]{geometry}

% or like this:

\usepackage[hmargin=1.25in,vmargin=1in]{geometry}

又比如，需要设定周围的边距一致为 1.25 英寸，可以用更简单的语法：

\usepackage[margin=1.25in]{geometry}

对于书籍等双面文档，习惯上奇数页右边、偶数页左边留出较多的页边距，而书脊一侧的奇数页左边、偶数页右边页边距较少。可以这样设定：

\usepackage[inner=1in,outer=1.25in]{geometry}

geometry 宏包本身也能够修改纸张大小、页眉页脚高度、边注宽度等等参数。更详细的用法不再赘述，感兴趣的用户可查阅 geometry 宏包的帮助文档。

### 页面内容垂直对齐

LaTeX默认将页面内容在垂直方向分散对齐。对于有大量图表的文档，许多时候想要做到排版匀称的页面很困难，垂直分散对齐会造成某些页面的垂直间距过宽，还可能报大量的Underfull \vbox 消息。

LaTeX还提供了另一种策略：将页面内容向顶部对齐，给底部留出高度不一的空白。在导言区或者适合的位置使用以下命令开启顶部对齐的效果：

\raggedbottom

相反地， \flushbottom 命令用于设置成默认的分散对齐。

### 使用颜色

LaTeX原生不支持颜色，它依赖 color 宏包或者 xcolor 宏包，给 PDF 输出生成颜色的特殊指令。

调用 color 或 xcolor 宏包后，就可以用如下命令切换颜色：

\color[⟨color-mode⟩]{⟨code⟩}

\color{⟨color-name⟩}

颜色的表达方式有两种，其一是使用色彩模型和色彩代码，代码用 0 ∼ 1 的数字代表成分的比例。 color 宏包支持 rgb、 cmyk 和 gray 模型， xcolor 支持更多的模型如 hsb 等，示例如图4.34所示。

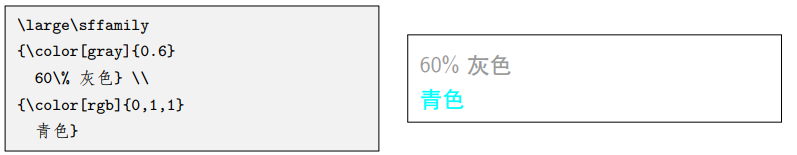


图4.34 颜色设置

color 宏包仅定义了 8 种颜色名称， xcolor 补充了一些，总共有 19 种，详见图4.35。

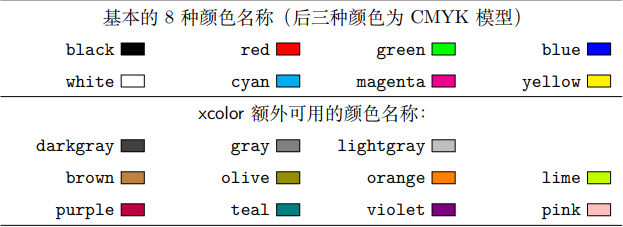


图4.35 颜色模型

xcolor 还支持将颜色通过表达式混合或互补，如图4.36所示。

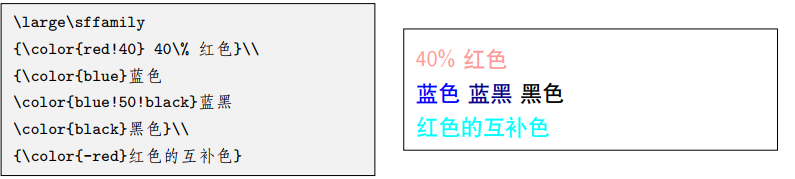


图4.36 颜色使用示例

原始的 \color 命令类似于字体命令 \bfseries，它使之后排版的内容全部变成指定的颜色，所以直接使用时通常要加花括号分组。 color / xcolor 宏包都定义了一些方便用户使用的带颜色元素。输入带颜色的文本可以用类似 \textbf 的命令：

\textcolor[⟨color-mode⟩]{⟨code⟩}{⟨text⟩}

\textcolor{⟨color-name⟩}{⟨text⟩}

以下命令构造一个带背景色的盒子， ⟨material⟩ 为盒子中的内容：

\colorbox[⟨color-mode⟩]{⟨code⟩}{⟨material⟩}

\colorbox{⟨color-name⟩}{⟨material⟩}

以下命令构造一个带背景色和有色边框的盒子， ⟨fcode⟩ 或 ⟨fcolor-name⟩ 用于设置边框颜色：

\fcolorbox[⟨color-mode⟩]{⟨fcode⟩}{⟨code⟩}{⟨material⟩}

\fcolorbox{⟨fcolor-name⟩}{⟨color-name⟩}{⟨material⟩}

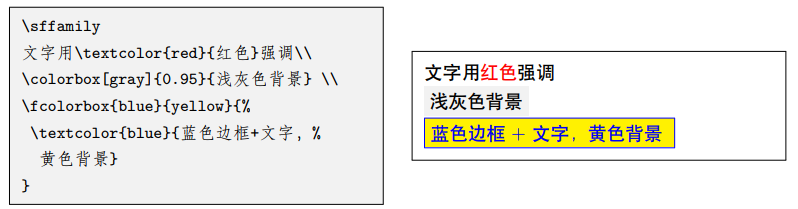


图4.37 使用示例

### 添加超链接

PDF 文档格式是现今最流行的电子文档格式，而电子文档最实用的需求之一就是链接功能。LaTeX中实现这一功能的是 hyperref 宏包。hyperref 宏包提供了直接书写超链接的命令，用于在 PDF 中生成 URL：

\url{⟨url⟩}

\nolinkurl{⟨url⟩}

\url 和 \nolinkurl 都生成可以点击的 URL，区别是前者有彩色，后者没有。在 \url 命令中作为参数的 URL 里， 可直接输入如 %、 & 这样的特殊符号。也可以像网页一样，把一段文字赋予其“超链接”的作用：

\href{⟨url⟩}{⟨text⟩}



图4.38 添加超链接示例

使用 hyperref 宏包后，文档中所有的引用、参考文献、索引等等都转换为超链接。用户也可对某个 \label 命令定义的标签 ⟨label⟩ 作超链接（注意这里的 ⟨label⟩ 虽然是可选参数的形式，但通常是必填的）：

\hyperref[⟨label⟩]{⟨text⟩}

默认的超链接在文字外边加上一个带颜色的边框（在打印 PDF 时边框不会打印），可指定colorlinks 参数修改为将文字本身加上颜色，或修改 pdfborder 参数调整边框宽度以“去掉”边框； hidelinks 参数则令超链接既不变色也不加边框。

\hypersetup{hidelinks}

% or:

\hypersetup{pdfborder={0 0 0}}

## LaTeX编译错误排查技巧

### 报错排查示例

当用排版引擎编译 LaTeX代码时，命令行的窗口（终端）会显示大量信息（TEXworks等编辑器会有一个区域显示这些信息）。当编译过程中出现错误时，信息将会停止在出错的地方，等待接下来的操作。

比如说下面这个明显出错的例子：

\documentclass{article}

\begin{document}

Test \LaTeX{} and it’s friends.

\end{document}

编译过程中遇到这个错误将会停顿下来，提示错误，并等待用户输入指令：

! Undefined control sequence.

l.3 Test \LaTeX

{} and it’s friends.

这种错误信息分两部分，前一部分提示了错误的信息，后一部分指出了错误发生的行号，以及通过错落的文字告知发生错误的命令所在位置。如上错误显示 \LaTeX位置发生了错误，错误信息是“未定义的控制序列”，意思是 \LaTeX是 TEX 编译器无法识别的一个命令，很显然是把 \LaTeX的大小写写错了。

### 处理方式

出现错误时，编译过程将暂停，等待用户输入命令。可以直接敲回车跳过当前的错误，继续编译，相当于丢掉了写错的命令，将 “Test and it’s friends.’’ 排版出来。但这个例子过于简单，有些复杂的代码中，有可能会由于一个小问题导致一连串的错误。此时可以选择按 S/R/Q选择跳过接下来的所有错误，或者按 X 直接退出编译，将源代码中的错误修改后重新编译。

### 常见的LaTeX错误及处理

! Undefined control sequences.

使用了未定义的命令。拼写错误是原因之一，如把 \LaTeX写作 \LaTEX这样。也有可能是没有调用某个宏包，但用了该宏包定义的命令。

! LaTeXerror: Environment ... undefined.

使用了未定义的环境。

! Missing $ inserted.

缺少数学环境的符号 $。多由于将数学符号用在公式之外而导致。

Runaway argument?

! Paragraph ended before ... was complete.

! File ended while scanning definition/use of ...

这两个错误主要是由于漏写了包裹命令参数的花括号，导致识别参数时出现错误。许多编辑器的括号配对功能有助于检查和消除这类错误。这类错误还有可能是由于前一次编译中断导致 .aux 等辅助文件不完整，再次编译读入不完整的文件产生错误。解决办法是删除辅助文件并重新编译。

! Extra alignment tab has been changed to \cr.

! Misplaced \noalign.

两个错误信息都与表格有关。前者的字面意义是“一行中使用的列分隔符 & 太多”，有时可能确实是 & 的个数和列格式不匹配，但多数情况是漏掉了行尾的 \\ 命令。

后者常出现于漏掉了行尾的 \\ 命令而接着使用 \hline 命令画横线的时候。

! LaTeXError: Lonely \item--perhaps a missing list environment.

! LaTeXError: Something’s wrong--perhaps a missing \item.

两个错误信息都与列表环境和 \item 命令有关。前者意味着在没有使用列表环境的情况下用了 \item；后者则相反，是在列表环境中漏了 \item。

! I can’t find file ‘...’.

! LaTeXError: File ‘...’ not found.

两个错误都意味着缺少文件。如果使用 \input 或者 \include 命令添加文件，出现上述错误的原因当然是文件不存在或者文件名不对；如果错误提示里的文件名带 .cls 或者 .sty 扩展名，那么很显然，是因为没有安装所需的宏包或文档类。

! LaTeXError: Missing \begin{document}.

字面上是缺少 \begin{document}，实际上往往是由于在 \begin{document} 之前（导言区）输入了文字或某些命令。

! LaTeXError: Cam be used only in preamble.

与上一条相反，由于将必须用于导言区的命令在 \begin{document} 之后使用而产生。

! LaTeXError: \begin{...} on input line ... ended by \end{...}.

环境首尾不匹配。比如 \begin{enumerate} 用了 \end{itemize} 结尾。或者也可能是由于漏写了 \begin 或者 \end 命令。

! LaTeXError: Option clash for package ‘...’.

以不同选项重复调用宏包造成冲突。有可能是因为其它宏包内部事先调用了这个宏包，用户再次带选项调用而导致冲突。解决问题的办法是去掉重复调用的宏包。如果宏包允许的话，尽量使用其定义的命令改变设置，减少宏包选项的使用。

! LaTeXError: Command ... already defined,

or name \end... illegal ...

使用 \newcommand 或 \newenvironment 定义已有的命令/环境时产生的错误。如果自己确实作了定义，可考虑用 \renewcommand 或 \renewenvironment 定义；如果是宏包定义的命令产生了这个错误，则属于隐性的宏包冲突。相比之前的 “Option Clash”，隐性宏包冲突是更难以解决的问题，对各种宏包不熟悉的用户，尤其是使用模板的用户而言，往往难以下手。用户可尝试查找引起冲突的宏包的帮助手册。详尽的手册里通常会告知用户这个宏包应当在某个宏包的前面/后面调用，或者不能与某个宏包一起调用。如果是模板调用了大量宏包导致冲突，可联系模板的作者解决。

! LaTeXError: Unknown option ‘...’ for package ‘...’.

调用宏包时指定了不能被其识别的选项。此时应该查找宏包的帮助文档来解决问题。

! Package ‘...’ error: ...

宏包或文档类自定义的错误，由于不正确地使用宏包里的命令而导致。此时应该查找宏包的帮助文档来解决问题。

# 联合开发及迭代规则

## 联合开发与迭代

### 联合开发模式

为方便不同部门之间修订手册，（尤其是部分SIP芯片，手册部分内容由专人撰写，出现多人维护同一份手册的情况）目前已在公司gitlab上进行托管手册的源码，手册源码管理权限由专人负责，通过邀请的方式，将相关手册编写人员拉到组内，授予开发者权限。联合开发必须要遵循以下规则，严格执行：

* 规则一：任何修改必须创建分支（不得直接基于主分支修改，包括源码管理者），在分支上进行修改，修改完成之后，在提交合并请求；
* 规则二：任何合并请求，必须填写修改原因和修改内容位置，并且在手册源码中按规则添加修订历史记录；
* 规则三：审核人员在审核合并分支时，必须严格按照手册的规范检查，如有不合规范的地方，截图通知对应的修改人员，按手册规范进行修改，满足要求后再并入主分支；
* 规则四：发布前需更新版本号，必须由管理者按版本升级规则进行更新和管理版本号；
* 规则五：主分支必须永远保留最新版本的手册，可以直接对外发布。

## 发布方式及规则

手册的发布，需要保证gitlab上手册工程的主分支里，始终保留一份最新版本编译好的手册，统一发布在公司论坛或官网。需专人定期从gitlab更新到官网及论坛提供客户下载和使用。

1. 附录

在手册编写中，存在很多重复出现的东西，将这部分内容提取出来，做成宏及样式模版，使用时直接调用即可。

* 1. 常用宏
     1. 颜色宏定义

首页中存在非黑色字体及表格颜色，在preamble.tex中定义的颜色，设置了RGB的比重，需要使用时，直接调用即可。

%%%%%%%%%%%%颜色%%%%%%%%%%%%%%%%

\definecolor{ZLGblue}{RGB}{51,51,153}

\definecolor{ZLGblue1}{RGB}{51,102,255}

\definecolor{ZLGgreen}{RGB}{20,167,201}

\definecolor{history}{RGB}{255,204,153}

* + 1. 表格相关宏定义

在preamble.tex中定义了表格所使用元素的参数，详见程序清单5.1，直接调用即可。

程序清单5.1 表格相关元素

\heavyrulewidth = 0.08em %\toprule线宽

\lightrulewidth = 0.08em %\midrule线宽

\belowrulesep = 0pt %线后间距

\belowbottomsep = 100pt % 表后空距离

\abovetopsep = 10pt %表前空距离

\newcommand\subsec[1]{\subsection{#1}

\setlength\LTpre{0pt} %长表格与上文的距离

\setlength\LTpost{-40pt} %长表格与下文的距离

\arrayrulecolor{black}

\setlength\LTpre{5pt} %长表格与上文的距离

\setlength\LTpost{-20pt} %长表格与下文的距离

}

* + 1. 文档布局相关

在preamble.tex中定义了标题间距相关、正文缩进参数、图表间距参数等，详见程序清单5.2，直接调用即可。

程序清单5.2 文档布局参数

\usepackage[explicit,newlinetospace]{titlesec}

%\usepackage{xcolor,lipsum}

\newcommand{\yiji}{\fontsize{15.96}{15.96}\selectfont} %一级标题15.96pt，行距15.96pt

\newcommand{\erji}{\fontsize{13.98}{13.98}\selectfont} %二级标题13.98pt，行距13.98pt

\newcommand{\sanji}{\fontsize{12}{12}\selectfont} %二级标题12pt，行距12pt

\colorlet{sectcolor}{white}

\usepackage{framed}

\setlength{\hangindent}{6em}

\setlength\leftskip{0pt} %正文缩进

\setlength\headsep{30pt} %页眉底部到页主体顶部的距离

\setlength\footskip{-0.5cm}

\setlength{\textheight}{24cm}

\setlength{\textwidth}{17.19cm} %设置文本部分的高度和宽度

\setlength{\parindent}{2em} %段前缩进量

\setlength{\parskip}{4pt} %段落间距

\linespread{1.6}\selectfont %正文单倍行距

1. 常用模版样式

在LaTeX中绘制表格，对用户手册中出现较多的几种表格，在preamble.tex中定义了表格的样式，使用时，直接调用该样式即可开始表格内容的编写，无需自己绘制表格样式。

* 1. 寄存器概述表

寄存器概述表基本上每个外设都需要使用，在preamble.tex中定义了寄存器概述表格的样式，详见程序清单5.3，以RCC寄存器为例调用该表格后，生成的表格详见图5.1。

程序清单5.3 寄存器概述表

\newcommand\regOverview[2]

{\begin{center}

\captionsetup{

margin=0cm,

labelsep=period,

labelfont=bf,

labelfont=hei,

textfont=bf,

textfont=hei,}

\setlength{\abovecaptionskip}{0pt}%

\setlength{\belowcaptionskip}{0pt}

\small

\begin{longtable}{|p{50pt}|p{80pt}|p{200pt - 60pt}|p{20pt + 40pt}|p{99pt}|}

\caption{#1} \\

\hline\hline

\rowcolor{mygray}

\begin{minipage}{1.5cm} \vspace{0.2cm} {\bfseries Offset} \vspace{0.2cm}\end{minipage} &

\bfseries Acronym &

\bfseries Register Name &

\bfseries Reset &

\bfseries Section \\ \hline\hline

\endfirsthead

\hline\hline

\rowcolor{mygray}

\begin{minipage}{1.5cm} \vspace{0.2cm} {\bfseries Offset} \vspace{0.2cm}\end{minipage} &

\bfseries Acronym &

\bfseries Register Name &

\bfseries Reset &

\bfseries Section \\\hline\hline

\endhead

\hline\hline

\endfoot

#2

\end{longtable}

\end{center}}

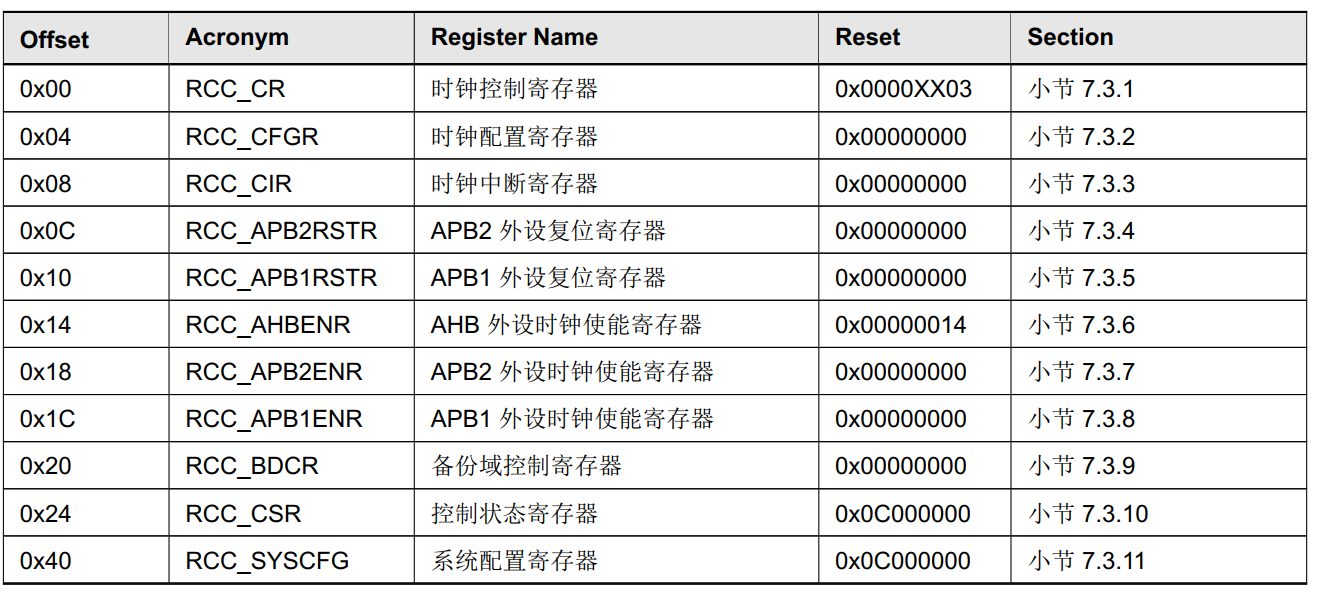


图5.1 寄存器概述表

* 1. 寄存器描述表

寄存器描述表基本上每个寄存器都需要使用，在preamble.tex中定义了寄存器描述表格的样式，详见程序清单5.4，以RCC寄存器为例调用该表格后，生成的表格详见图5.2。

程序清单5.4 寄存器描述表

\newcommand\regDes[1]

{

% \rowcolors{2}{table\_old\_color}{table\_even\_color} %enable colors

\begin{center}

\begin{longtable\*}[c]{|m{40pt}|m{60pt}|m{40pt}|m{40pt}|m{243pt}|}

\hline\hline

\rowcolor{mygray}

\begin{minipage}{1.5cm} \vspace{0.2cm} {\bfseries Bit} \vspace{0.2cm}\end{minipage} &

\bfseries Field & \bfseries Type & \bfseries Reset & \bfseries Description \\

\hline\hline

\endhead

\hline

\endlastfoot

#1

\end{longtable\*}

\end{center}

}

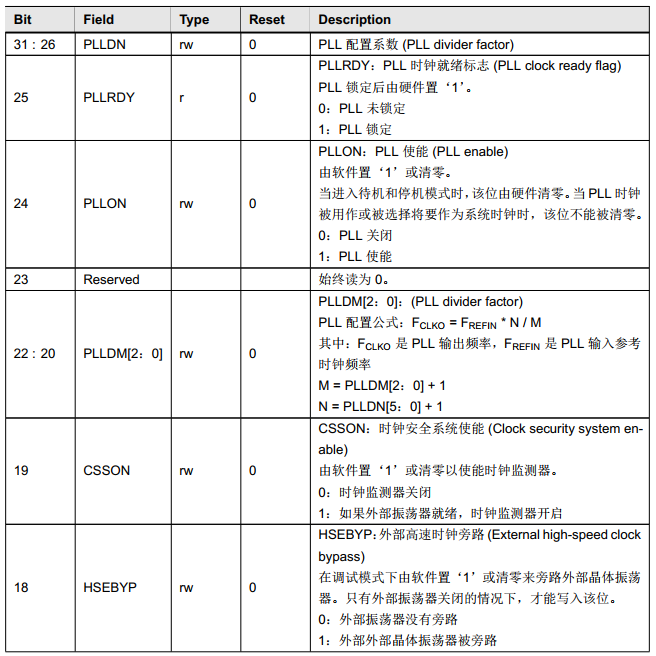


图5.2 寄存器描述表