

|  |  |
| --- | --- |
| **文档编号** |  |
| **版本号** | **V0.0.1** |
| **归档部门** | **系统部** |

**版本管理**

**【项目产品管理】**

**李 阳**

**二〇二〇年一月**

**变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 日期 | 作者/修改者 | 描述 | 版本号 |
| 1 | 2019-10-23 | 李阳 | 创建 | 0.0.1 |
| 3 |  |  |  |  |

目录

[1 引言 4](#_Toc30332936)

[2 版本管理工具 4](#_Toc30332937)

[3 使用规则 4](#_Toc30332938)

[3.1 概要流程 4](#_Toc30332939)

[3.2 版本细则 4](#_Toc30332940)

[3.2.1软件 4](#_Toc30332941)

[3.2.2 硬件 5](#_Toc30332942)

[3.2.3 结构 5](#_Toc30332943)

# 1 引言

待补充

# 2 版本管理工具

采用github作为版本工具（后期换为公司私有git服务器）

Git地址：

# 3 使用规则

## 3.1 概要流程

1.版本管理员分配git账号给对应人员，根据需求分配权限

2.在项目初始阶段，管理员在母仓库建立项目仓库（软件目录、硬件目录、结构目录和软件固件目录）

3.项目对应开发者或者测试人员，fork对应仓库到自己github，再克隆到自己的工作站中进行开发。

4.当产品进行到里程碑节点时，申请合并版本到母仓库。

5.此时管理员将develop分支下的内容合并到release分支下作为测试版本。

6.测试完成后，评估此版本是否可以发布，若可以则管理员将release分支下的内容提交到master分支下作为量产版本（稳定版本）。

## 3.2 版本细则

### 3.2.1软件

**1.目录结构：**

code // 代码源文件

tools // 对应配置软件，调试软件等附加软件

doc // 产品所需要或者涉及到的文档

**2.重要说明**

1.code中必须具有changelog日志。见changelog模板

2.code中的源代码版本按照MAJOR.MINOR.LITE(V0.0.0)方式打标签。MAJOR为稳定版本标识，MINOR和LITE由开发者分配，范围从0~255。Major变动，minor和lite清0。Minor变动，Lite清0。

3.tools中的开发的软件需有版本标识

4.doc中的产品相关文档（即除了芯片等参考手册等第三方文档）命名方式均采用“项目名\_文档名\_版本\_日期”作为文档名称。

5.软件固件仓库中包含“版本说明”文件，介绍版本内容。仓库中放置对应bin或者hex等相关文件，具体参阅“版本发布流程”文件

### 3.2.2 硬件

### 3.2.3 结构

文档格式要求：

1. 一级标题：宋体四号、加粗，格式数字加标题，如：“**1 引言**”
2. 二级标题：宋体四号、加粗，格式数字加标题，如：“**1.1 编写目的**”
3. 三级标题：宋体小四、加粗，格式数字加标题，如：“**1.1.1 采集原理**”
4. 正文：宋体11号，常规，单倍行距，首行缩进2个字符。