

AMetal 代码仓库使用说明

AMetal

V1.0.0 Date:2018/12/05

标准规范手册

类别	内容
关键词	AMetal、Git、TortoiseGit、版本控制
摘 要	本文档简述了 AMetal 代码仓库的使用、TortoiseGit 的使用以及版本控制等。

修订历史

版本	日期	原因
发布 1.0.0	2018/12/5	创建文档

目 录

1. AMetal 简介.....	1
2. 环境搭建	1
2.1 Git 与 Github	1
2.2 Git for windows	2
2.2.1 安装	2
2.2.2 运行	3
2.3 TortoiseGit.....	4
2.3.1 安装	4
2.3.2 中文语言包安装.....	5
2.3.3 软件设置.....	5
2.3.4 用户信息配置.....	8
3. 简单地使用版本库（无需 Github 账号）	9
3.1 克隆远端仓库	9
3.2 更新本地仓库	10
3.3 创建新的分支	12
3.4 切换分支.....	13
4. 代码贡献	13
4.1 Github 账号注册	13
4.2 生成 SSH 密钥	15
4.2.1 确认本地密钥.....	15
4.2.2 生成密钥信息.....	15
4.2.3 将 SSH 私钥转换为 PPK 格式	16
4.2.4 将公钥上传至 Github 服务器	18
4.3 获取用户仓库下 URL.....	20
4.4 添加新文件	21
4.5 提交修改.....	22
4.6 合并分支.....	23
4.7 消除冲突.....	24
4.8 查看日志信息及回退版本	27
4.9 推送至远端服务器	28
4.10 发起“Pull Request”请求	29
5. F&Q	32
5.1 创建.ssh 文件夹时，创建失败.....	32
5.2 在克隆以及提交部分，TortoiseGit 选项卡内无克隆及提交选项	32

1. AMetal 简介

AMetal 是芯片级的裸机软件包，定义了跨平台的通用接口（使得基于 AMetal 的应用程序可以和具体硬件完全分离，实现跨平台复用），并提供一系列驱动及常用的软件服务。

AMetal 开源于 Github，其开源网址为“<https://github.com/zlgopen/ametal>”，用户可直接通过此页面下载 AMetal 开源代码包。如图 1.1 所示，点击“Clone or download”，再点击“Download ZIP”即可下载开源代码包。

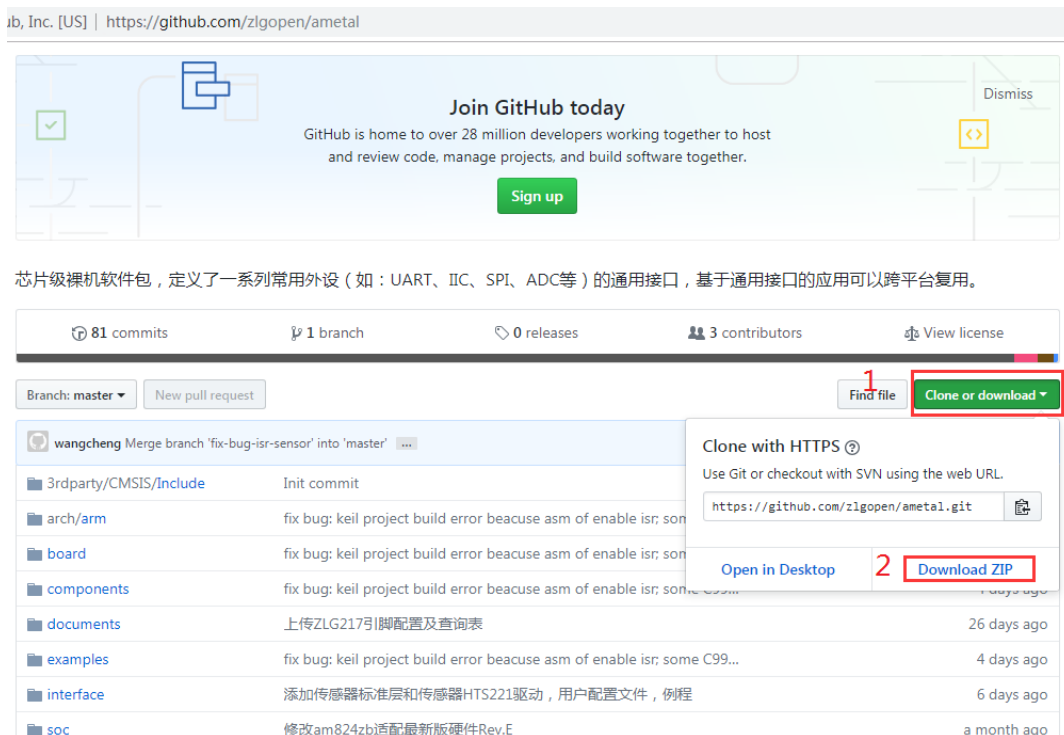


图 1.1: 在 Github 上下载 AMetal 代码包

下载解压后，在“ametal/documents”目录下有一系列文档，可帮助用户快速了解 AMetal 并使用 AMetal 进行项目开发。

由于 AMetal 开源项目在持续的更新，用户在使用 ZIP 方式下载时，有如下几处缺陷：

- 不易对代码进行维护；
- 不易具体的知道代码更新情况；
- 不易定位更新代码位置。

由于上述缺陷，本文将介绍一种维护代码的方法，并使用此方法对 AMetal 代码进维护。

2. 环境搭建

2.1 Git 与 Github

Git 是一个开源的分布式版本控制系统，可以有效、高速地完成从很小到非常大的项目版本管理；Github 是一个面向开源及私有软件的托管平台，且其只支持 Git 作为唯一的版本库格式进行管理。

Git 的原生管理工具是通过命令行进行操作，对于不熟悉的 Git 用户来说可能不太友好，而 TortoiseGit 为 Git 提供了一个便于操作的图形化界面，它可以方便地显示已修改的文件，提供可视化具有对比性的合并工具和差异显示工具，同时其比较合并工具对解决在冲突查找问题的帮助很大，相对命令行更加方便。

本文将介绍一种使用 Git 与 TortoiseGit 对代码进行管理的方式，由于其具有可以直接在本地进行更新、可获取更新信息、能定位更新文件等优点，使用其维护代码，能让代码维护工作变得非常简单。

2.2 Git for windows

Git for windows 是将 Git 版本管理工具在 windows 上的实现，提供了和 Linux 一样的 Git 命令行工具。具体下载页面可见“<https://git-scm.com/downloads>”，用户可根据操作系统、电脑位数等不同选择不同版本进安装，下载页面如图 2.2 所示。本文以 64 位 Win7 系统为例进行安装说明。

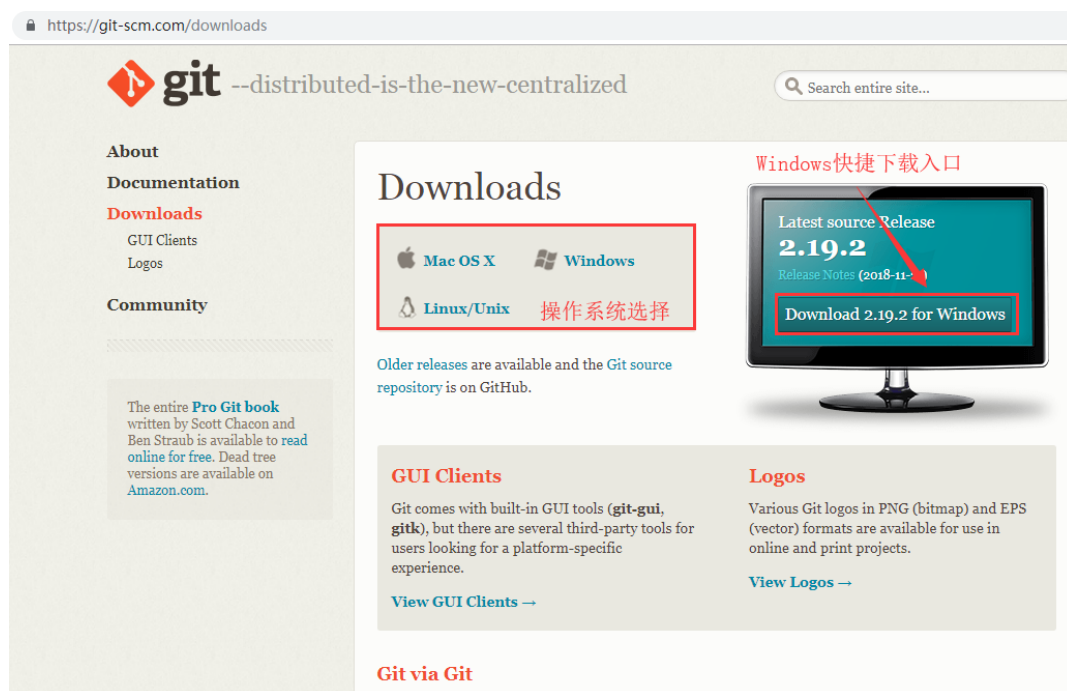


图 2.2: Git 下载页面

注解：截止本文编写完成之前，Git for windows 最新版本为 2.19.2，故本文选择使用该版本进行安装。在用户选择版本下载时，推荐选择不低于此版本的安装包进行下载，避免造成不兼容问题。

2.2.1 安装

下载 Git 安装包之后，右键点击安装包，选择“以管理员身份运行”，然后直接以默认设置安装即可。

2.2.2 运行

Git for windows 提供了 Git 的命令行操作，同时集成了大部分的 Linux 基础命令行工具，要使用这个工具，首先要启动运行的环境，在随意一个目录下鼠标右键单击，如图 2.3 所示，单击菜单中的“Git Bash Here”选项，将弹出 Git Bash 的操作控制台。



图 2.3: 选择打开 Git Bash

该操作控制台是一个类似 Linux 终端的界面，用户可在其中执行各种 Linux 常见的命令行命令，如 ls、cd、vim 等等，当然最主要的是执行 Git 命令，如图 2.4 所示。

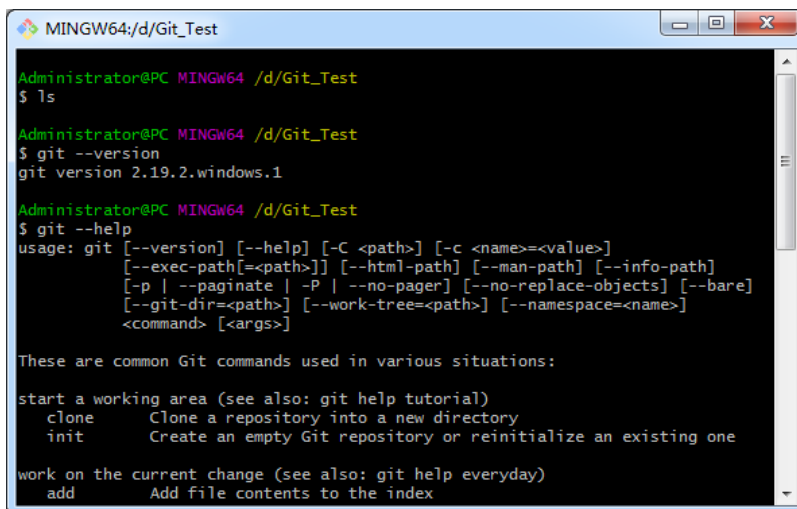


图 2.4: Git Bash 执行命令

2.3 TortoiseGit

用户可根据电脑的位数选择对应安装包以及对应语言包进行下载。具体下载页面可见“<https://tortoisegit.org/download/>”，下载页面如图 2.5 所示。

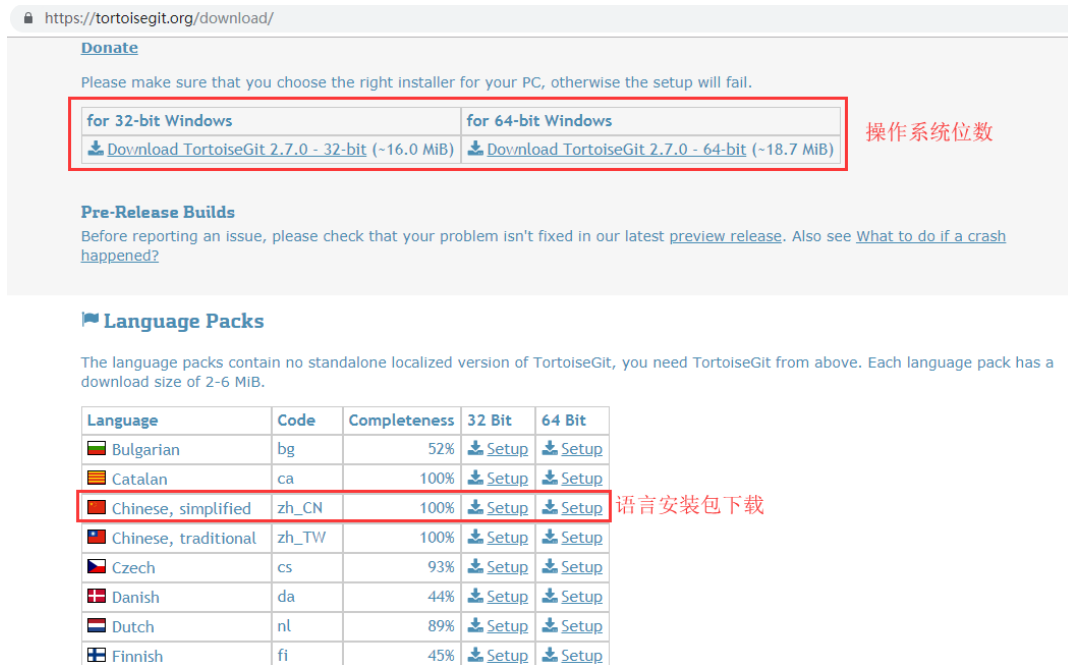


图 2.5: TortoiseGit 下载页面

注解：截止本文编写完成之前，TortoiseGit 最新版本为 2.7.0，故本文选择使用此版本进行安装。在用户选择版本下载时，推荐选择不低于此版本的安装包进行下载，避免造成不兼容问题。

2.3.1 安装

安装 TortoiseGit 时，点击 TortoiseGit 安装程序进行安装。

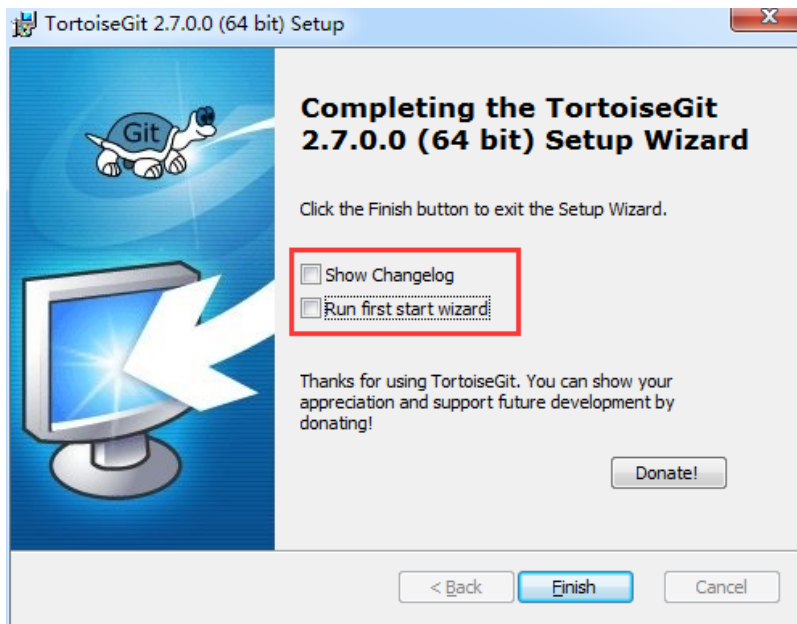


图 2.6: TortoiseGit 安装界面

在安装条进度界面完成时，会出现如 图 2.6 所示界面，取消“Run first start wizard”的选中框，然后点击“Finish”按钮完成安装。

2.3.2 中文语言包安装

双击 TortoiseGit-LanguagePack 进行安装，按照默认设置直接安装即可。

2.3.3 软件设置

安装 TortoiseGit 和语言包后，需要进行使用设置。首先在桌面空白处单击右键，如 图 2.7 所示操作。

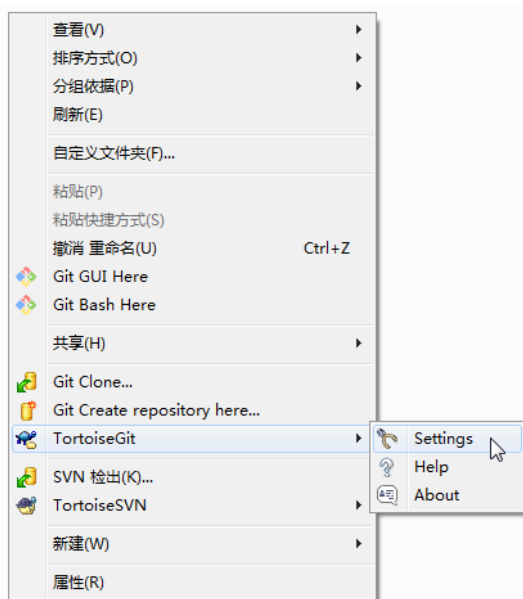


图 2.7: 进入 TortoiseGit 设置界面

接下来会出现设置主界面，如 图 2.8 所示，单击“Re-run First Start Wizard”，进入语言标准规范手册

设置界面，在这里可以选择中文和英文，如 图 2.9 所示。

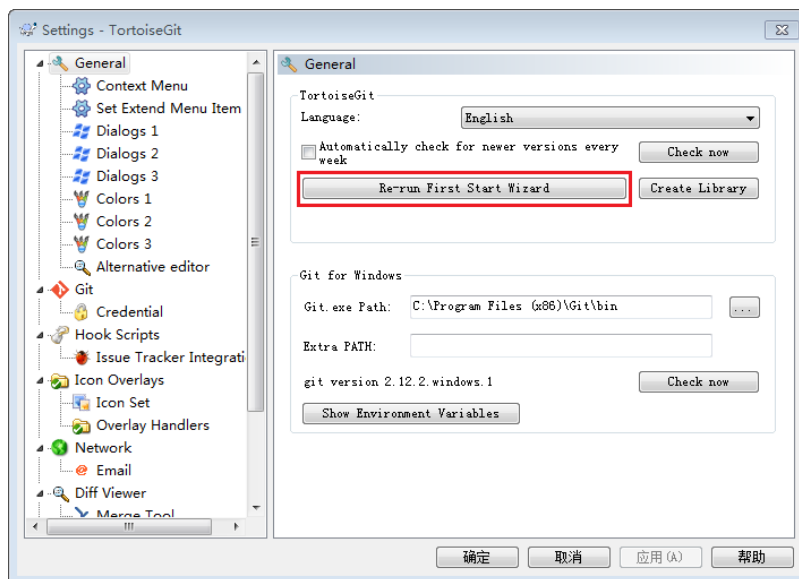


图 2.8: TortoiseGit 通用设置

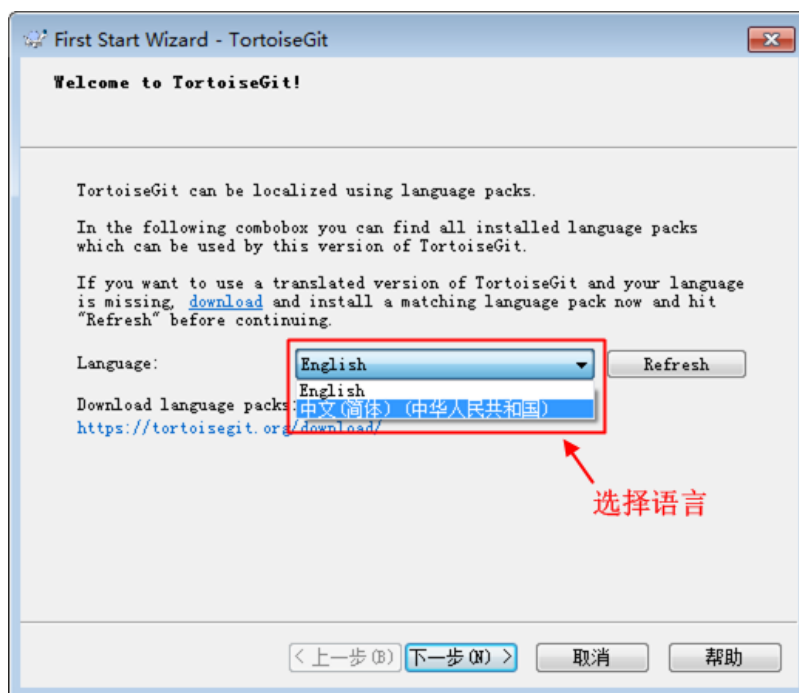


图 2.9: TortoiseGit 语言选择界面

完成语言设置后，单击“下一步”进入欢迎界面，如 图 2.10 所示。继续单击“下一步”，设置 Git 的路径，请按照 图 2.11 选择设置 git 路径，设置好 Git 路径后，单击“Check now”按钮可以显示当前 Git 的版本。



图 2.10: TortoiseGit 欢迎界面

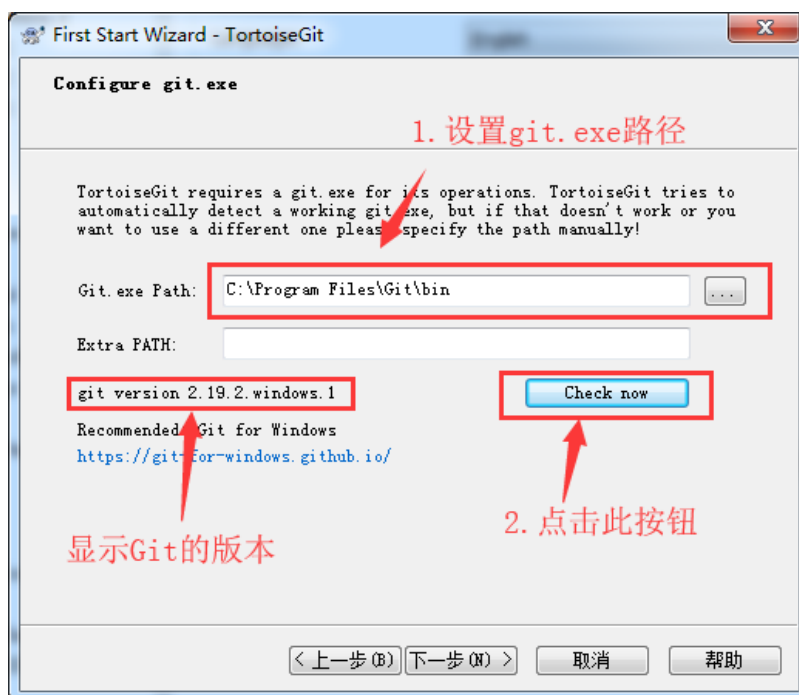


图 2.11: TortoiseGit 设置 Git 路径

单击“下一步”，进入如图 2.12 所示的设置用户名和邮箱界面，请填入后再单击“下一步”。

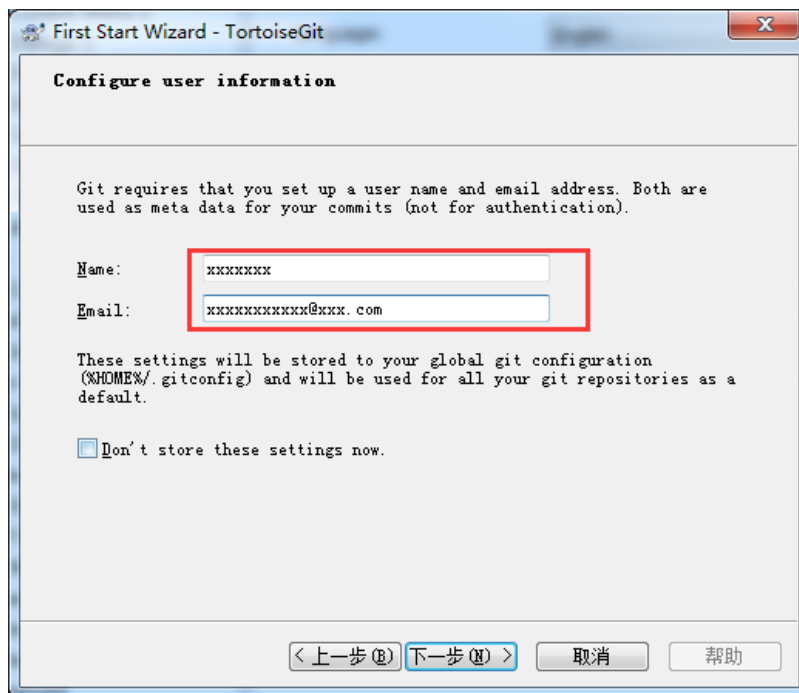


图 2.12: TortoiseGit 设置用户名及邮箱

接下来会出现身份验证和凭据存储界面，这里暂时先不进行相关设置，直接单击“完成”按钮即可。

2.3.4 用户信息配置

使用 TortoiseGit 配置，首先请在桌面空白处点击右键，选择“TortoiseGit->setting”，如图 2.13 所示。

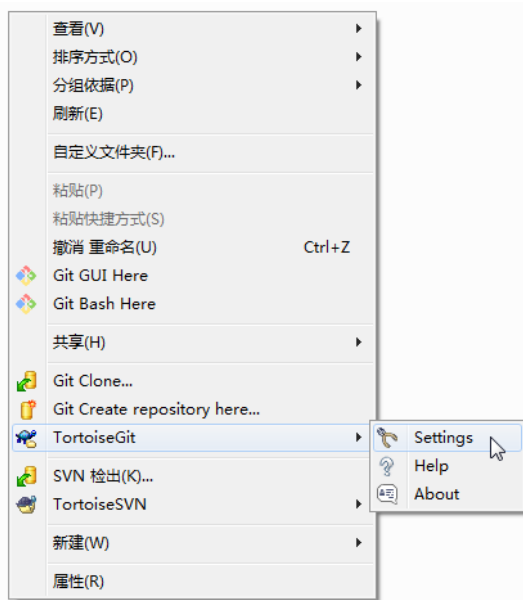


图 2.13: 打开 TortoiseGit 配置界面

然后进入 TortoiseGit 设置界面，如图 2.14 所示，点击被圈出的”Git”，进入配置对话框，按图中所示配置进行修改，修改完成后点击“确定”即可。

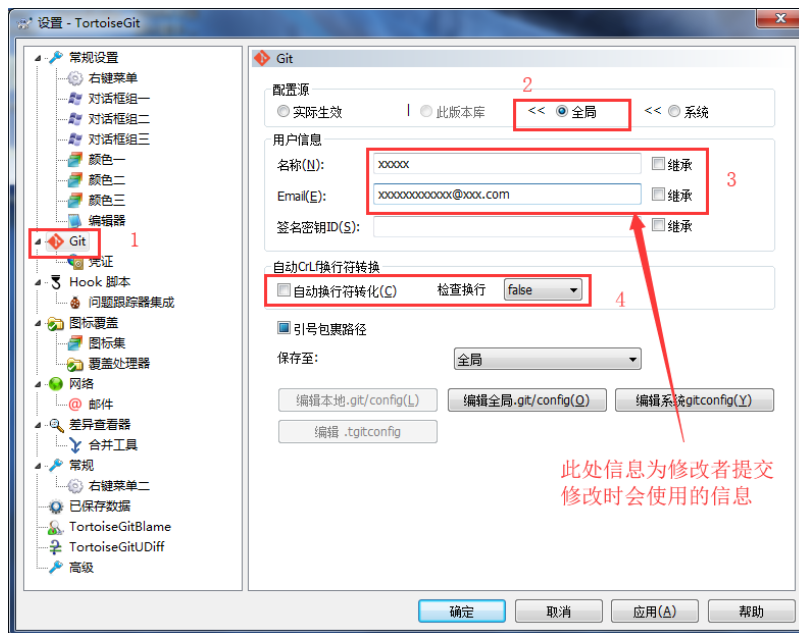


图 2.14: TortoiseGit 配置界面

3. 简单地使用版本库（无需 Github 账号）

AMetal 作为开源软件，始终欢迎广大开发者加入到 AMetal 开发的队伍中，向 AMetal 贡献自己的代码，丰富和完善 AMetal，使 AMetal 支持更多的 MCU、Board、组件、协议栈等。

本章节内容适用于仅使用版本库的用户。用户可以不关心 Github 的各项功能，也需使用命令行操作，直接按照文档说明即可使用版本库。

3.1 克隆远端仓库

在本地工作目录下空白处单击右键，选择“TortoiseGit -> 克隆”，会弹出如图 3.15 所示克隆对话框。请按图中红色文字的标注，将“https://github.com/zlgopen/amental.git”填入 URL 文本框以及本地工作目录，然后单击“确定”按钮。

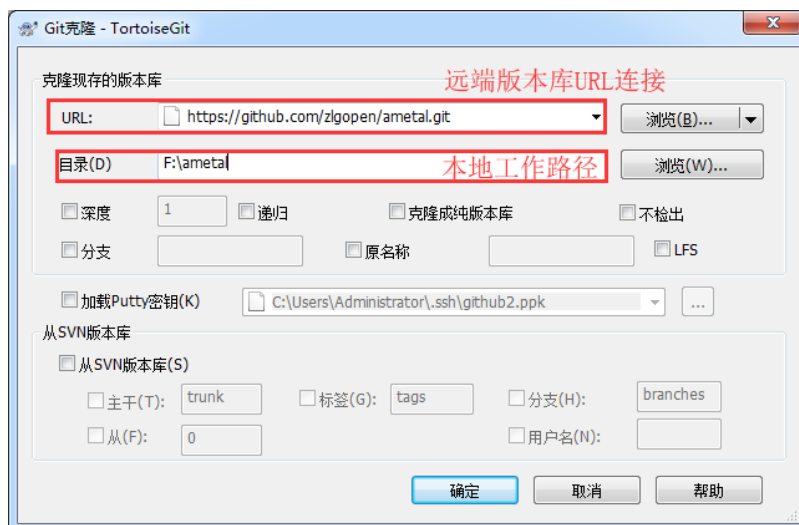


图 3.15: TortoiseGit 克隆界面

接下来会弹出一个 Git 命令进度对话框，如 图 3.16 所示，在克隆过程中会显示克隆完成进度。

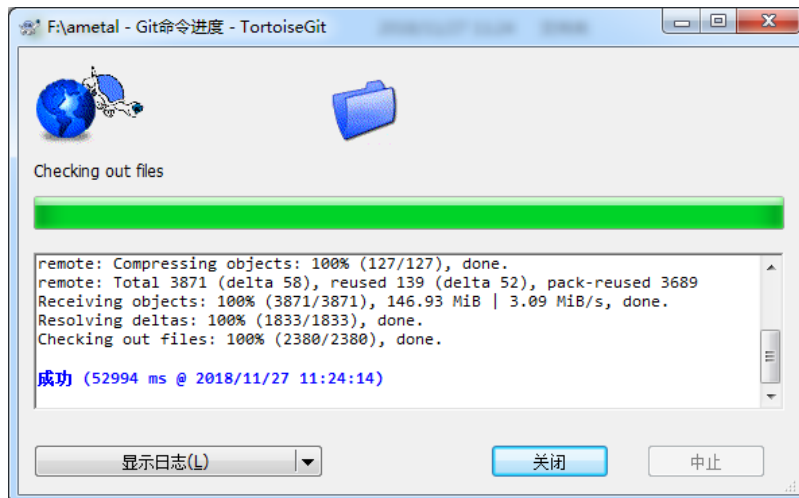


图 3.16: TortoiseGit 克隆进度界面

克隆完成之后，会出现如 图 3.17 所示界面，表示克隆版本库成功，点击“关闭”即可。

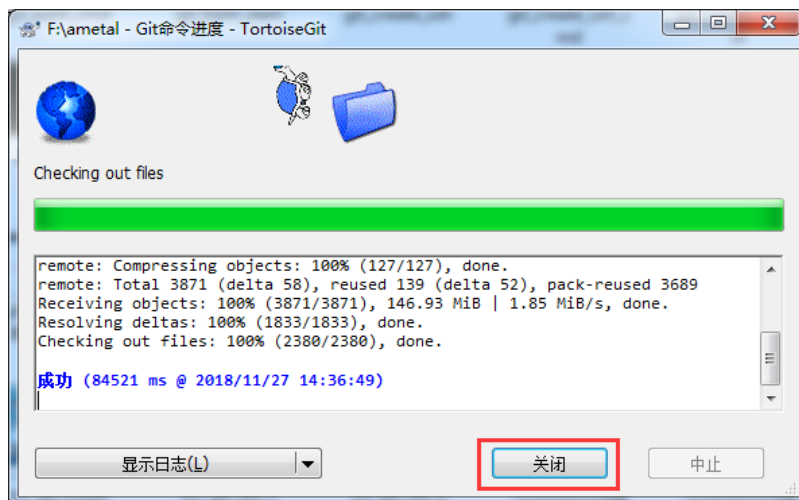


图 3.17: TortoiseGit 克隆完成界面

3.2 更新本地仓库

远端服务器上的 AMetal 版本库会持续进行更新，开发人员在完成 BUG 修复、代码优化、新的 MCU 型号支持的添加等工作后，会将版本库更新至 Github，当 Github 上的版本库先于用户本地版本库，若用户要使用最新的版本库，则需要对本地版本库进行拉取更新。

在本地工作目录下，在空白处单击右键，选择“TortoiseGit -> 拉取”，会弹出如 图 3.18 所示界面。

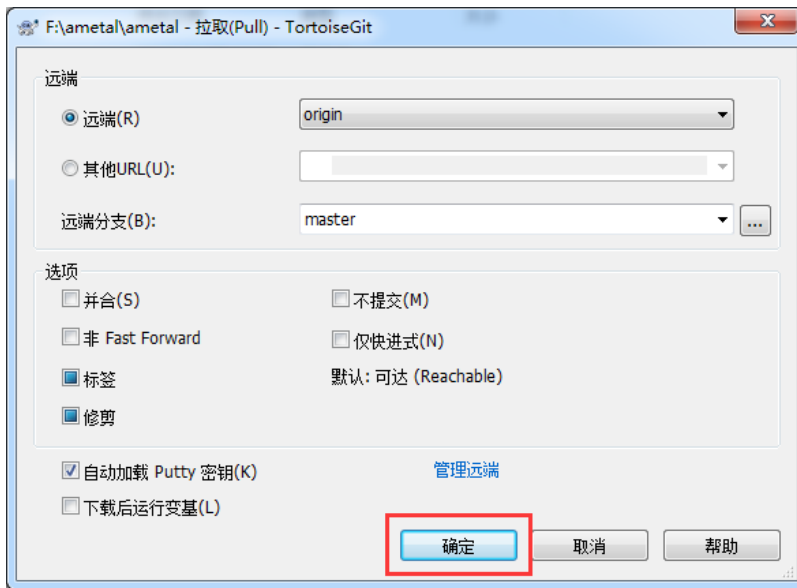


图 3.18: TortoiseGit 拉取界面

点击“确定”，会弹出一个拉取进度界面，如图 3.19 所示。

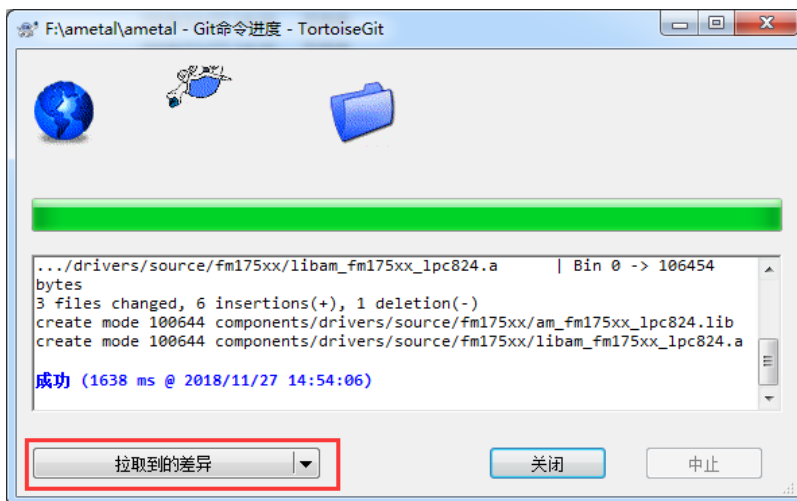


图 3.19: TortoiseGit 拉取进度界面

若点击“拉取到的差异”按钮，用户可以查看本地版本库与远端版本库的差异，也就是远端版本库更新/修改的内容，如图 3.20 所示。

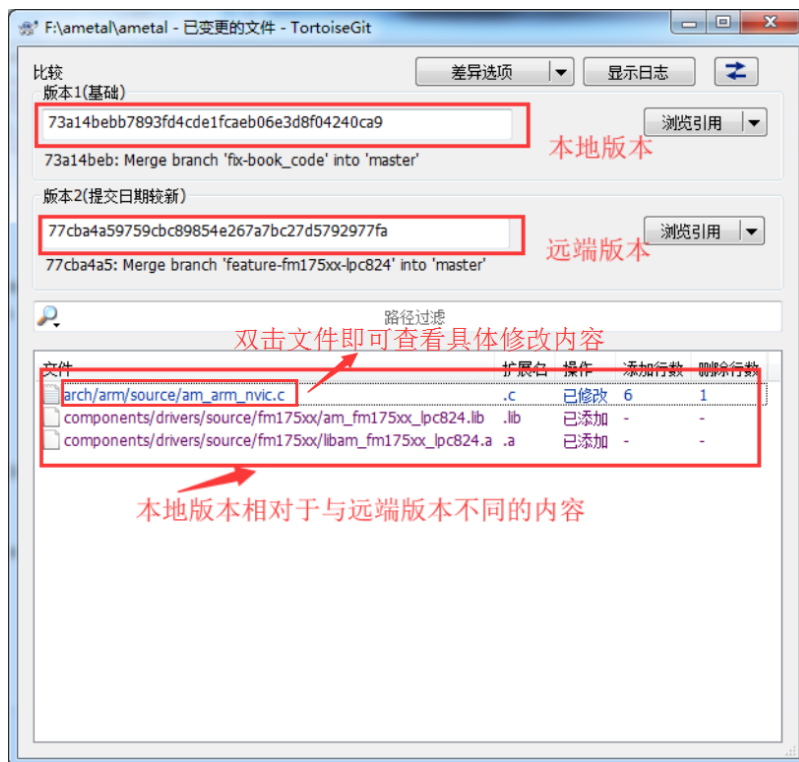


图 3.20: TortoiseGit 拉取差异查看界面

3.3 创建新的分支

通常为了防止用户误操作版本库造成版本库更新时造成冲突（冲突将在后面小节进行介绍），建议在版本库上建立分支对其进行使用。

在本地工作目录下，在空白处单击右键，选择“TortoiseGit -> 创建分支”，即可打开分支创建框，如 图 3.21 所示，输入分支名（用户可自由输入），勾选切换到新的分支，点击确定。完成操作后，用户就工作在以新克隆的版本库为基础的新分支上，在此分支修改、增加文件等操作不会影响到从远端克隆下来的版本库。

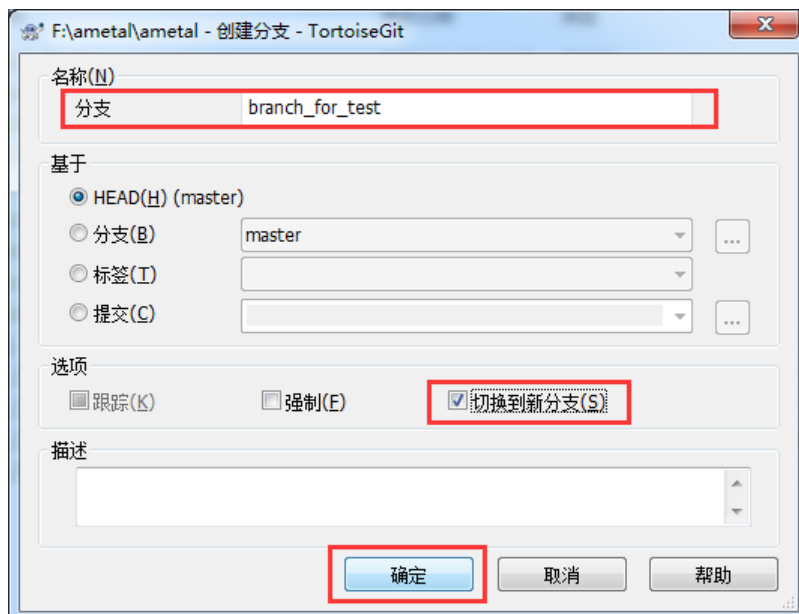


图 3.21: TortoiseGit 创建分支界面

3.4 切换分支

在本地工作目录存在多个分支的情况下，可以通过切换分支操作来在不同的分支下进行工作。

在本地工作目录下，在空白处单击右键，选择“TortoiseGit -> 切换/检出”，会弹出一个切换/检出框，如图 3.21 所示，用户可在此框内选择需要切换分支，然后点击“确定”，完成分支切换操作。

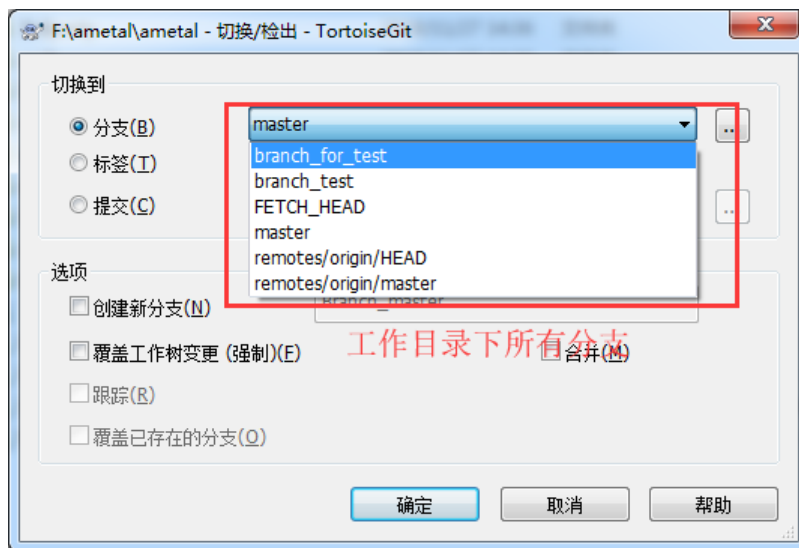


图 3.22: TortoiseGit 分支切换界面

4. 代码贡献

在本地工作时，若开发者编写了比较优秀的代码或者在使用 AMetal 过程中发现了某些错误并对其进行修改，开发者可以将添加或者修改的代码上传至版本库，向 AMetal 贡献代码。

本文以在 AMetal 版本库中添加测试功能为例，测试功能代码为新建立的 3 个文件，假设测试功能在测试分支被验证成功，则需要将包含这三个文件进行添加、提交、合并、推送等操作。

4.1 Github 账号注册

由于权限问题，用户无法直接将本地的代码推送至远端官方版本库中，若需要将代码上传至远端服务器的官方版本库，则需要使用到 Github 中的 Fork、Pull Request 等指令。

首先，进行 Github 账号注册，在浏览器中输入“<https://github.com/login>”开始访问 github。如图 4.23 所示会出现登录界面。

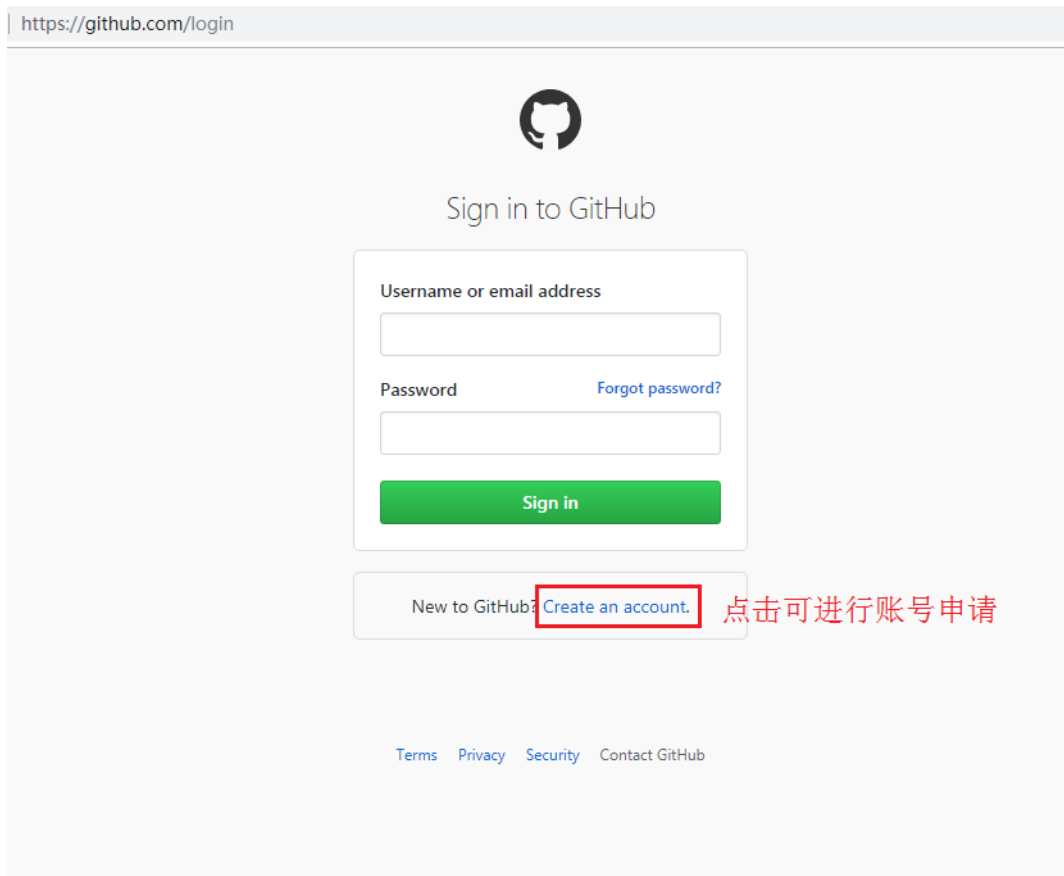


图 4.23: Github 登录界面

点击“Create an account”注册 Github 账户，会出现如 图 4.24 所示界面，按照步骤完成账号注册，若已经注册 Github 账户，直接输入用户名和密码后即可登录。

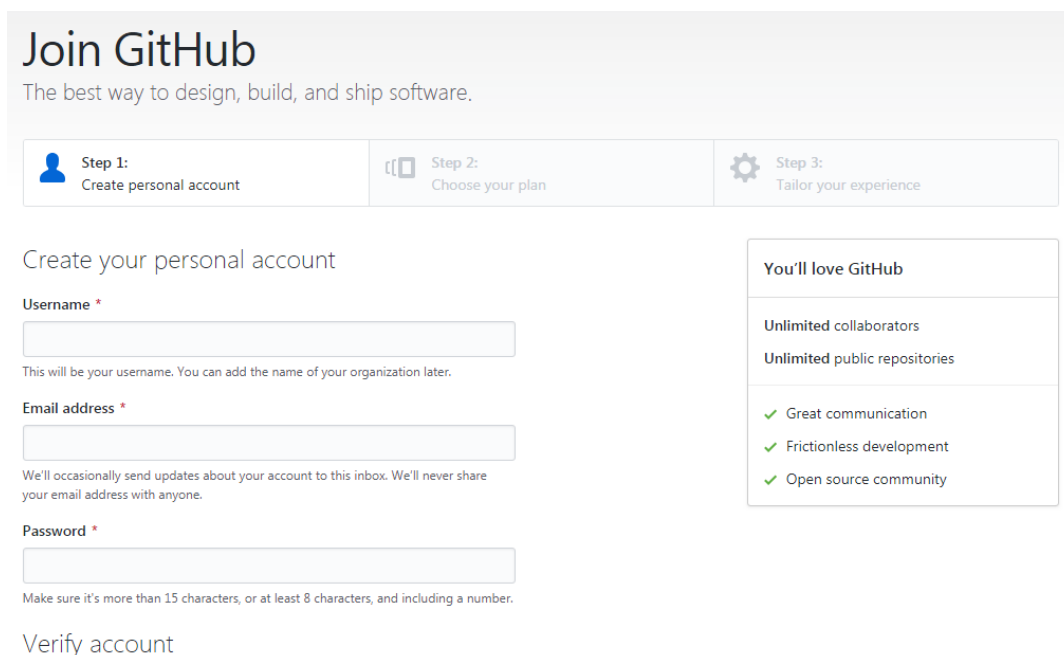


图 4.24: Github 用户注册页面

4.2 生成 SSH 密钥

此节操作用户可自由选择是否跳过。若不进行此操作，Git 使用 https 协议进行工作，在每次推送至远端时都需要输入密码进行验证；若进行此操作，Git 使用 SSH 密钥，可以在每次推送至远端时省去每次输入密码的过程。

4.2.1 确认本地密钥

SSH 密钥默认存储在账户的主目录下的 ~/.ssh 目录中，如：C:\Users\Administrator\.ssh\，查看是否包含 id_sra 和 id_rsa.pub，有 .pub 后缀的文件就是公钥，另一个文件则是私钥。如果该路径已经存在这两个文件，则跳过 4.2.2 小节。

4.2.2 生成密钥信息

在 .ssh 目录下空白处点击右键，选择 “Git Bash Here”，如 图 4.25 所示（若此目录不存在，则手动创建）。

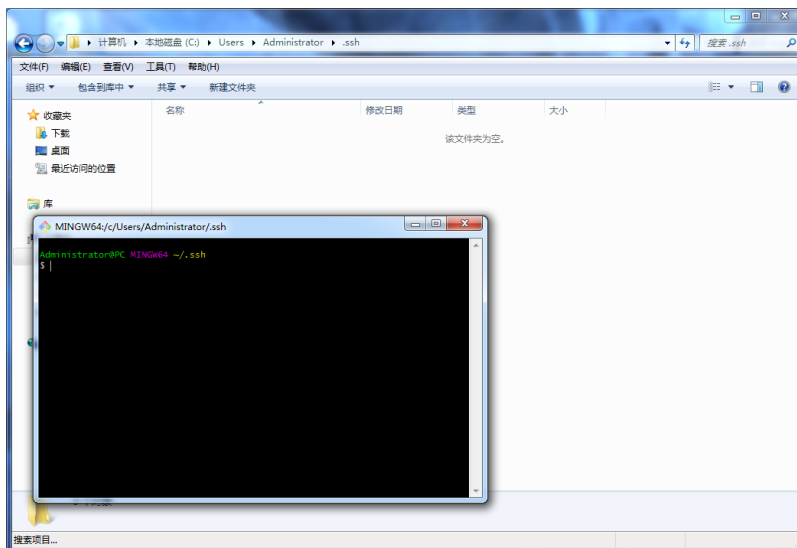


图 4.25: ssh 目录下打开 git bash

在 git bash 中输入 “ssh-keygen -t rsa -C your_email@xx.com” (此处的邮箱为 github 的邮箱), 然后执行，如 图 4.26 所示，命令窗口会提示用户选择保存密钥目录，点击回车，选择默认目录即可，然后命令行会提示输入密码，密码可输可不输，建议不输，直接回车即可。

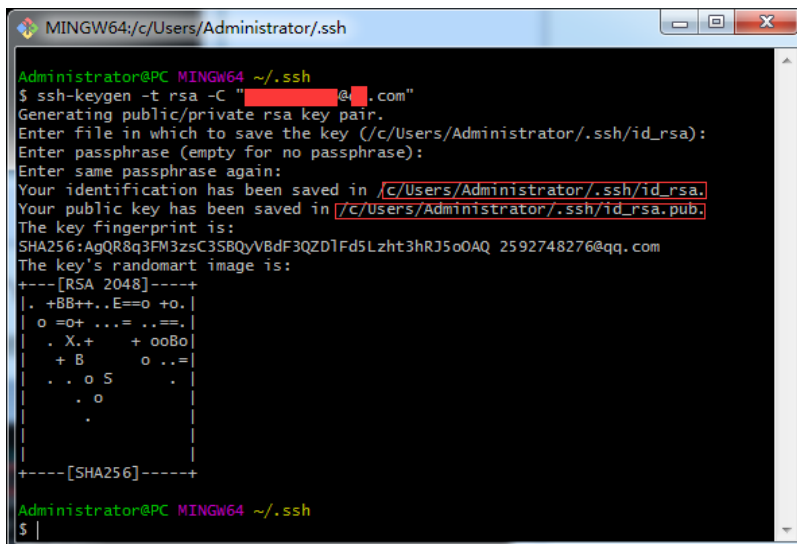


图 4.26: git bash 执行生成密钥命令

执行完成之后，会在.ssh 目录下生成密钥文件，如 图 4.27 所示。

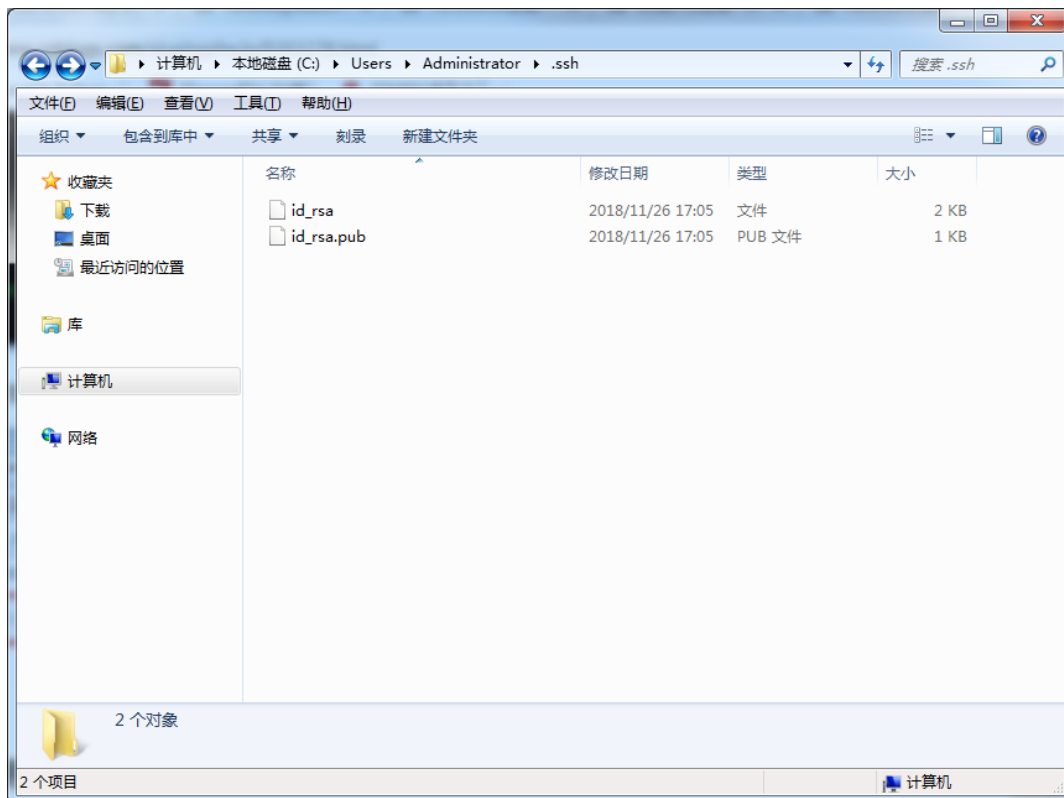


图 4.27: ssh 目录下生成的密钥文件

4.2.3 将 SSH 私钥转换为 PPK 格式

为了使用 TortoiseGit 执行 Git 的命令，我们需要将前面生成的 SSH 私钥使用 puttygen.exe 转换工具转换为 ppk 格式。

TortoiseGit 使用了 TortoisePlink.exe 来管理 SSH，但是 TortoisePlink.exe 只识别 ppk 格式的私钥，所以需要将公钥转换为 ppk 格式，转换用到的工具为 puttygen。

puttygen.exe 位于 TortoiseGit 的安装文件夹下 bin 文件夹中，启动 puttygen.exe，其界面

如 图 4.28 所示。接下来，请按 图 4.28 所示，选择菜单“Conversions”->“Import key”，在弹出的“加载私钥”对话框中，选择前面生成的 id_rsa 私钥文件。

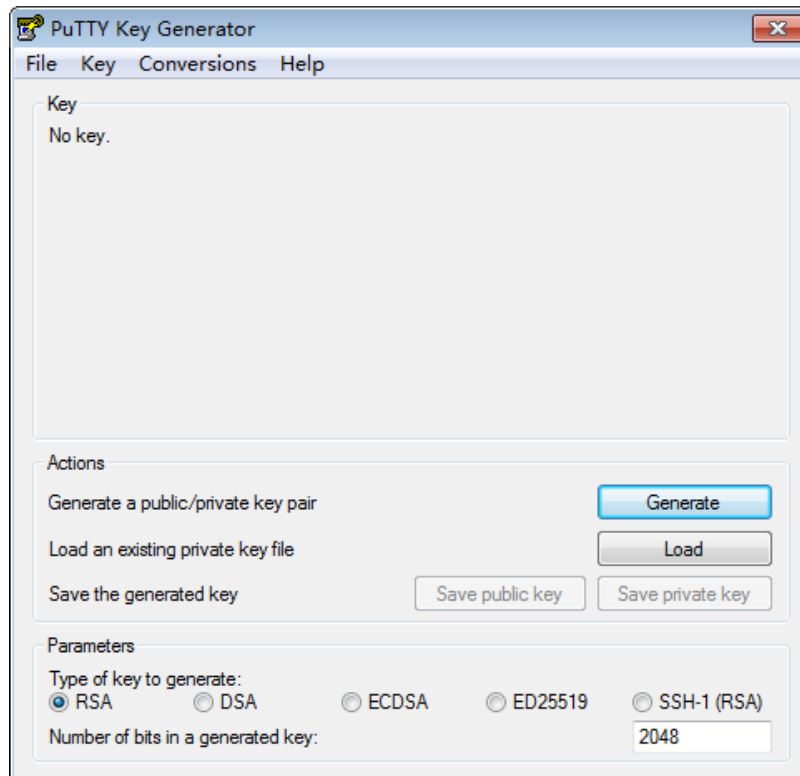


图 4.28: puttygen 主界面

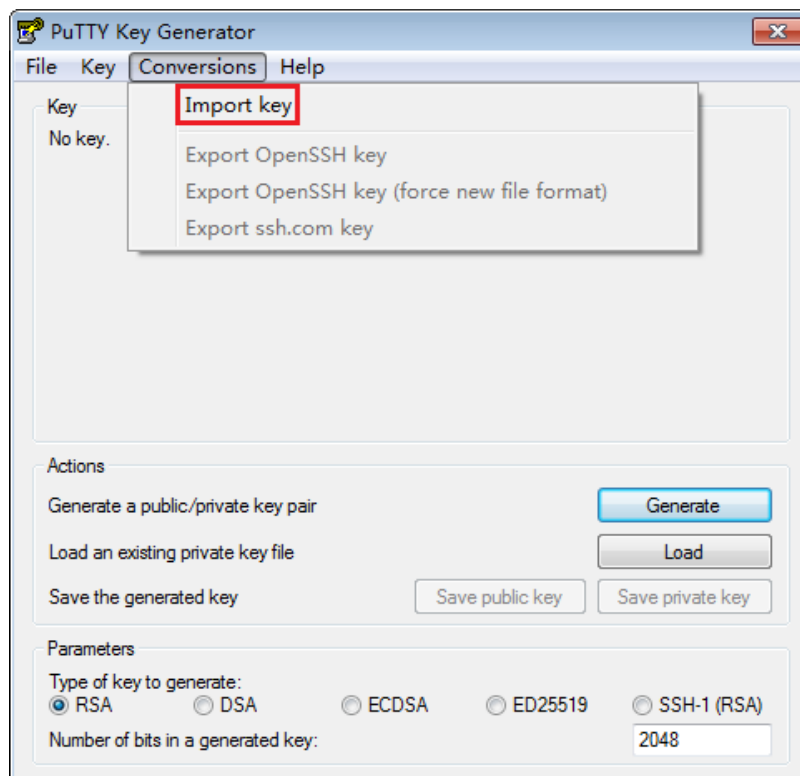


图 4.29: puttygen 导入密钥菜单

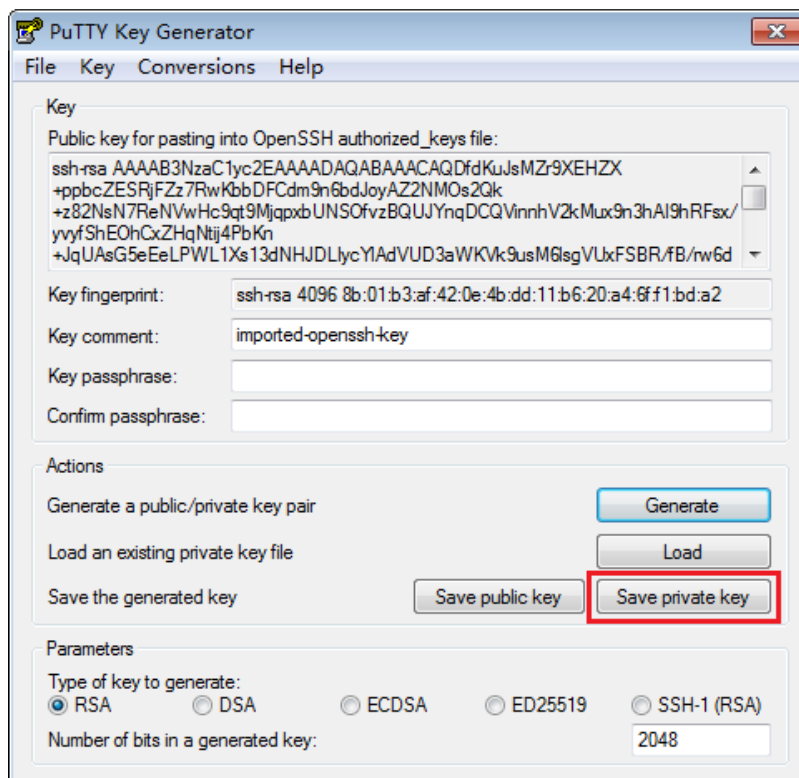


图 4.30: 私钥加载成功

加载私钥成功后，可以看到私钥加载成功后的界面，如图 4.30 所示，可以看到 key 区域已经有了内容。鼠标单击“保存私钥”按钮，将其命名为 github.ppk，并保存在.ssh 文件目录下。

4.2.4 将公钥上传至 Github 服务器

Git 直接使用 SSH 和 Github 上的 Git 版本库服务器通信，进行提交，获取，拉取等操作，因此需要将生成的 SSH 公钥上传并保存在 Github 服务器。

登录 Github 网页之后，点击右上角的头像，弹出下拉选项框，点击“Settings”按钮，如图 4.31 所示。

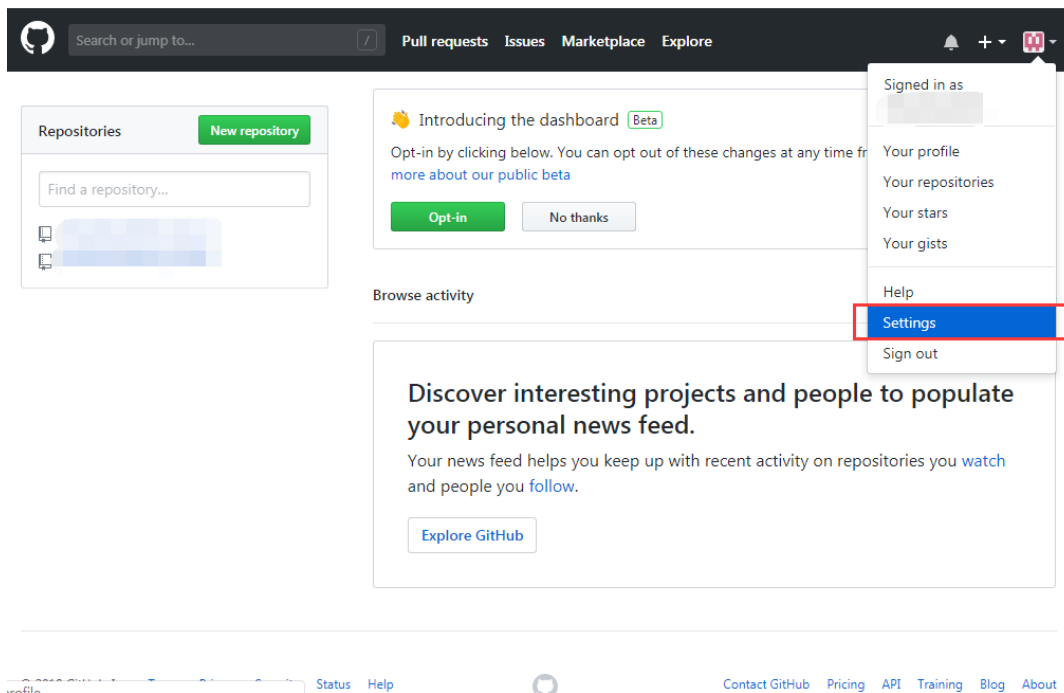


图 4.31: github 点击用户设置按钮

进入用户设置界面后，点击“SSH and GPG keys”按钮，进入公钥添加界面，再点击“New SSH Key”按钮，会进入如图 4.31 所示界面，在 Title 文本框内填入公钥标题（用户可随意自由填写），在 Key 文本框中填写上节中在 .ssh 目录下生成的 id_rsa.pub 文件内的内容（可用记事本打开），直接复制粘贴即可，点击“Add SSH key”完成添加。

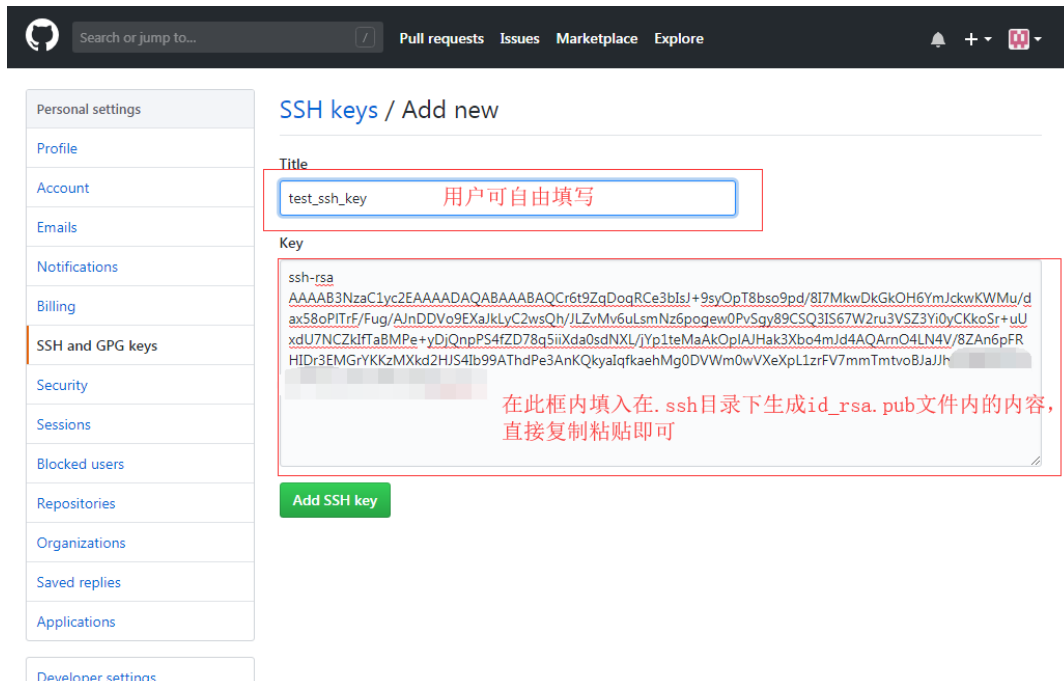


图 4.32: github 加载 SSH 公钥

4.3 获取用户仓库下 URL

进入 Github 首页，登录用户账号之后，搜索“ametal”，搜索结果如图 4.33 所示。点击“zlgopen/ametal”即可进入 AMetal 开源项目页面。

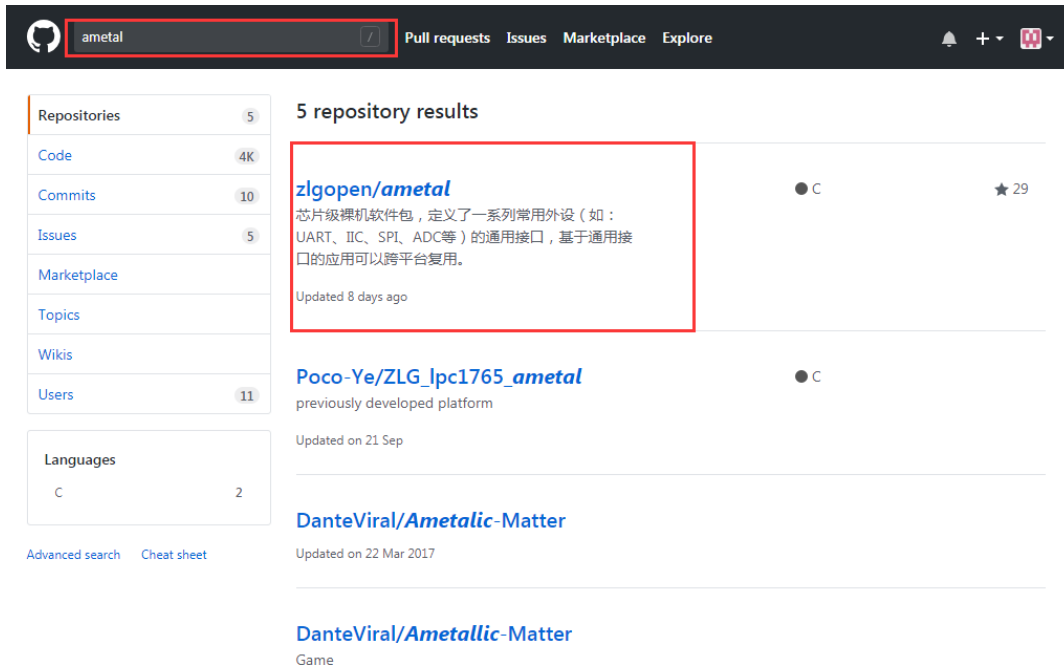


图 4.33: github 搜索页面

点击“Star”按钮，将项目标星，用户就可以在 Github 页面头像下“Your Stars”内快速找到该项目，点击“Fork”直接将该官方版本库克隆至用户自己的账户仓库下，如图 4.34 所示。

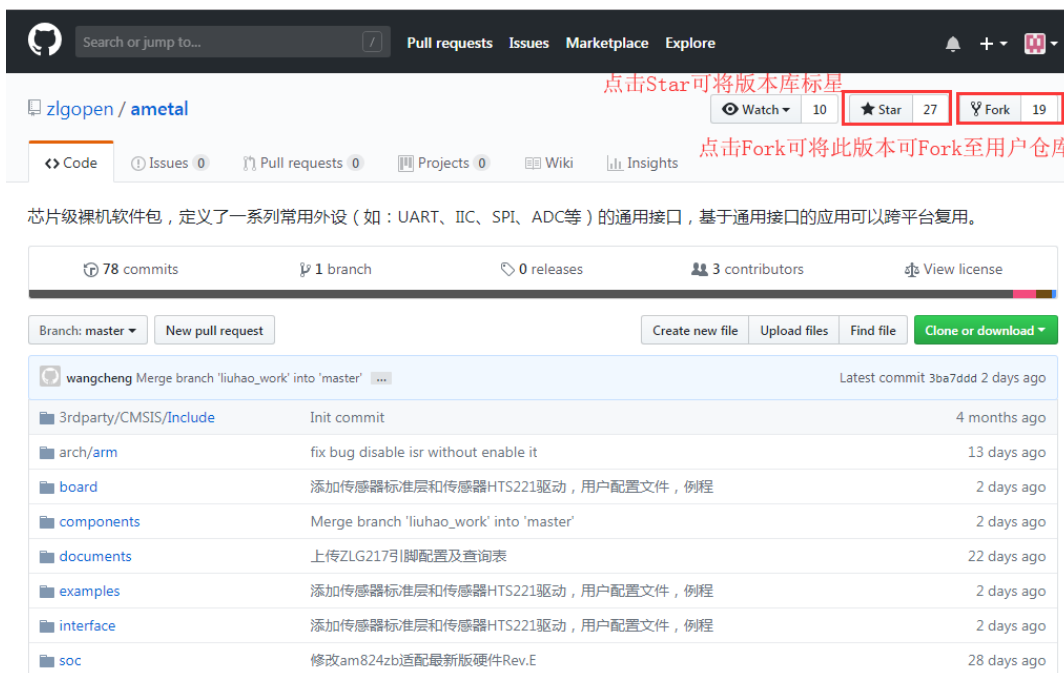


图 4.34: Github 中 AMetal 版本库页面

操作完成后，再次进入用户首页，点击选中 AMetal 项目，会进入如图 4.35 所示页面，点击“Clone or download”按钮，获取版本库的 URL 链接，直接复制即可。

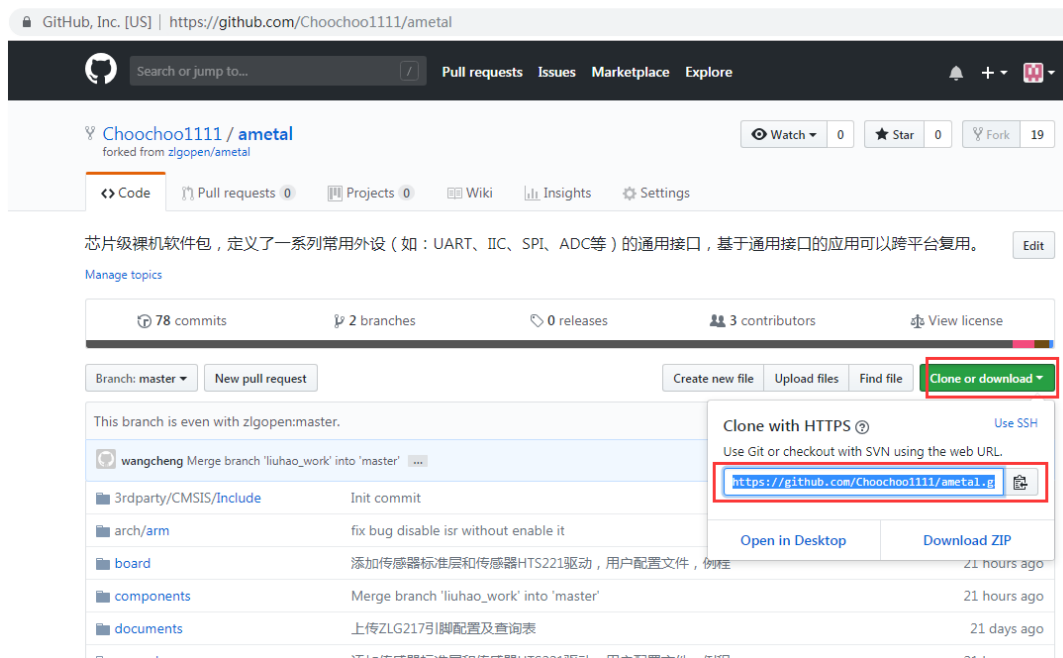


图 4.35: Git 用户账号仓库页面

注解：为了便于用户进行区分，将远端 AMetal 开源版本库简称为官方版本库，将用户克隆版本库简称为本地版本库，将用户 Fork 版本库简称为用户远端版本库。

4.4 添加新文件

如果在本地工作目录下添加了新文件，则需要将新文件添加进本地版本库进行管理。

在本地工作目录下，在空白处单击右键，选择“TortoiseGit -> 添加”，会弹出一个加入框，如图 4.36 所示，以图示为例，可看出当前目录下有三个文件未受版本控制，将其全部选中即可进行加入操作，用户在使用时可根据实际情况选择不同文件进行添加。

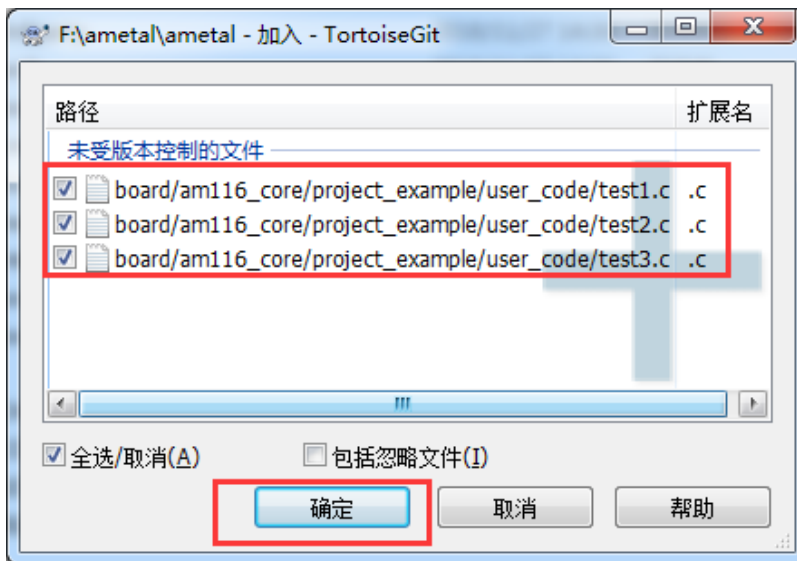


图 4.36: TortoiseGit 添加文件界面

点击“确定”即可完成添加。

4.5 提交修改

在工作目录下，若对文件进行修改、删除等操作之后，需要保存对文件的操作，则可以使用提交。

在本地工作目录下，在空白处单击右键，选择“TortoiseGit -> 提交”，会弹出一个提交框，如图 4.37 所示，变更列表中显示了本次操作相较于前一个保存的版本库所有操作过的文件。填写好本次修改的日志信息，并选择需要保存操作的文件。

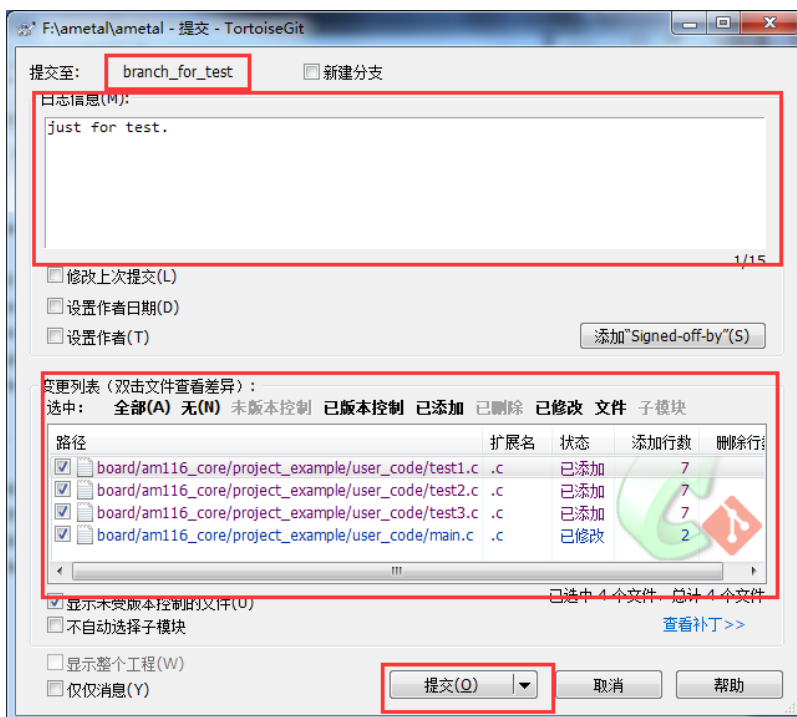


图 4.37: TortoiseGit 提交界面

点击“提交”后会弹出一个 Git 命令进度框，可在提示框内看到此次操作的相关信息。

如图 4.38 所示，点击“关闭”即可。

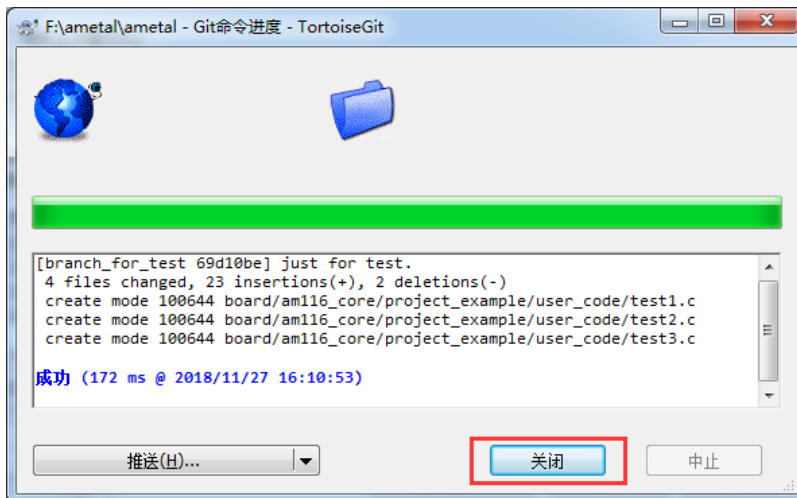


图 4.38: TortoiseGit 提交信息界面

4.6 合并分支

当存在为了验证某个不明确是否能成功的功能的情况时，可能会基于当前分支（工作分支）创建一个分支（测试分支）进行尝试，若此功能被验证为成功可行时，且需要将测试分支代码合并到工作分支时，就需要进行合并分支操作。

在本地工作目录下，在空白处单击右键，选择“TortoiseGit -> 合并”，会弹出一个合并框，如图 3.21 所示，选择要被合并的分支，并填写合并信息（可使用系统默认信息），然后点击“确定”，即可执行合并（默认合并是将所选择的分支合并到当前工作分支）。

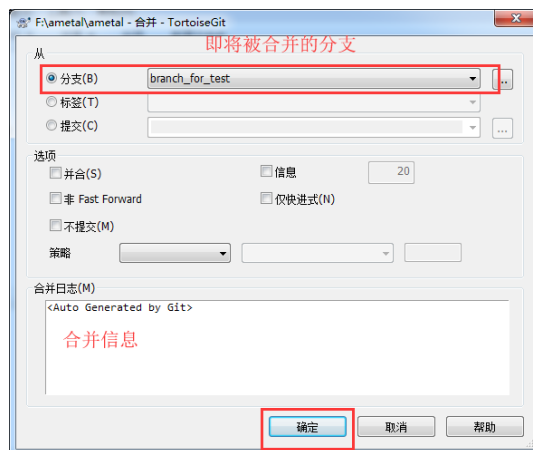


图 4.39: TortoiseGit 分支合并界面

点击“确定”后会出现如图 4.40 所示，若点击“删除分支”按钮，TortoiseGit 就会自动删除被合并过来的分支。



图 4.40: TortoiseGit 分支合并完成界面

注解: 若多个开发者同时开发同个项目时, 合并至主分支前, 可能会出现冲突, 用户需在合并时解决冲突, 冲突解决方法将在下节进行说明。

4.7 消除冲突

对于同一个版本库, 通常情况下会有多个开发者同时协同工作, 当多个开发者同时对一文件进行修改时, 在合并进主分支时, 就有可能发生冲突。通常在发生冲突的情况下, 需要先解决冲突, 然后在继续进行合并。

本文以 图 4.41 为例, 在进行合并时产生了冲突, 会弹出一个信息框提示用户在合并时产生了冲突并描述了解决办法, 在 Git 命令进度框上会提示了合并失败并显示了出现冲突的路径以及对应文件。

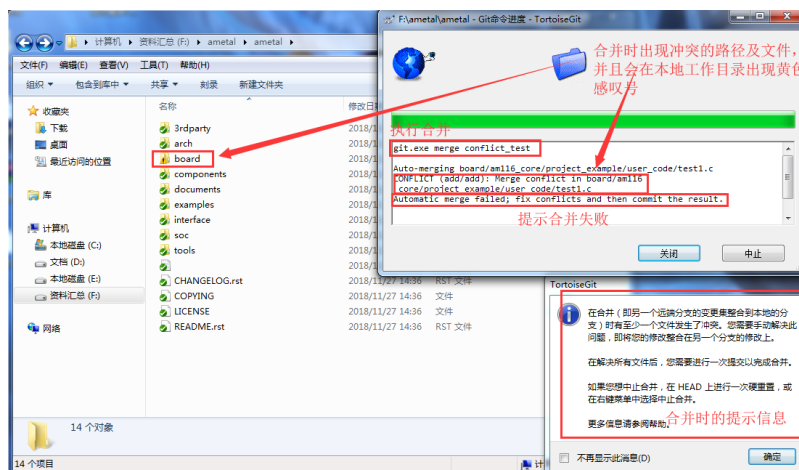


图 4.41: TortoiseGit 合并产生冲突

在关闭提示框之后, 右键点击空白工作区域, 选择“TortoiseGit -> 解决冲突”项, 如图 4.42 所示, 点击会弹出如图 4.43 所示提示框。

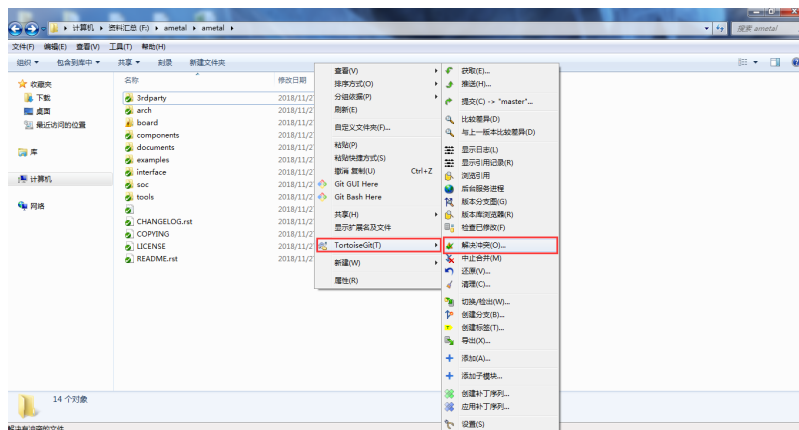


图 4.42: TortoiseGit 进入解决冲突界面

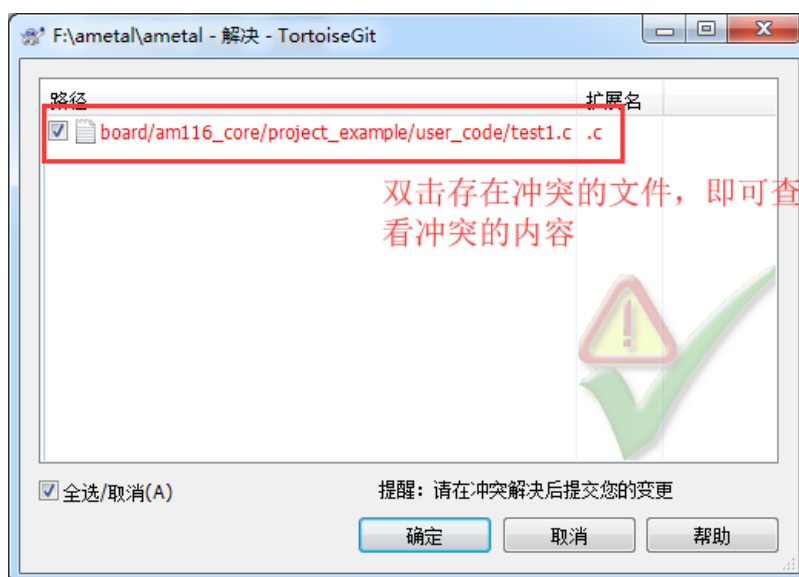


图 4.43: TortoiseGit 解决冲突界面

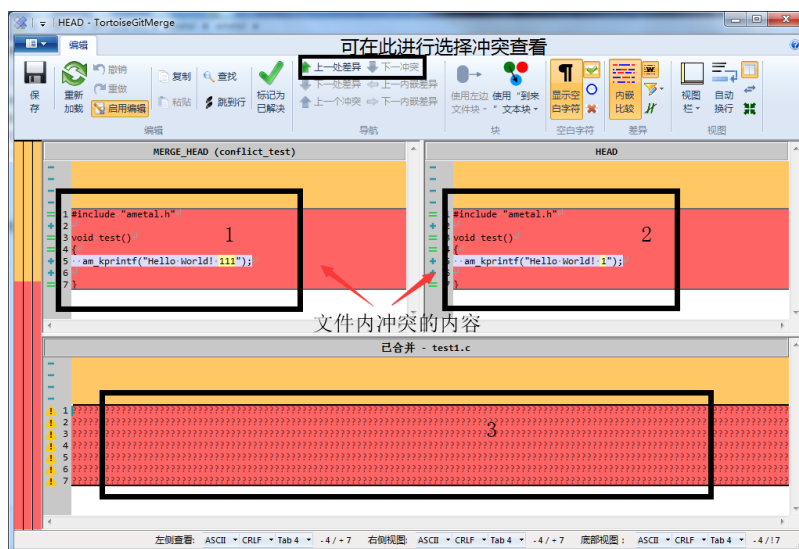


图 4.44: TortoiseGit 查看冲突内容

双击框内文件即可查看冲突文件内存在冲突的内容，如 图 4.44 所示，框 1 内代表的被合并的分支存在的冲突内容，框 2 内代表当前分支内与被合并分支存在冲突的内容，框 3 内

标准规范手册

代表用户进行冲突消除后保存的内容，由于目前还未解决冲突，所以框内全部显示为“?”。

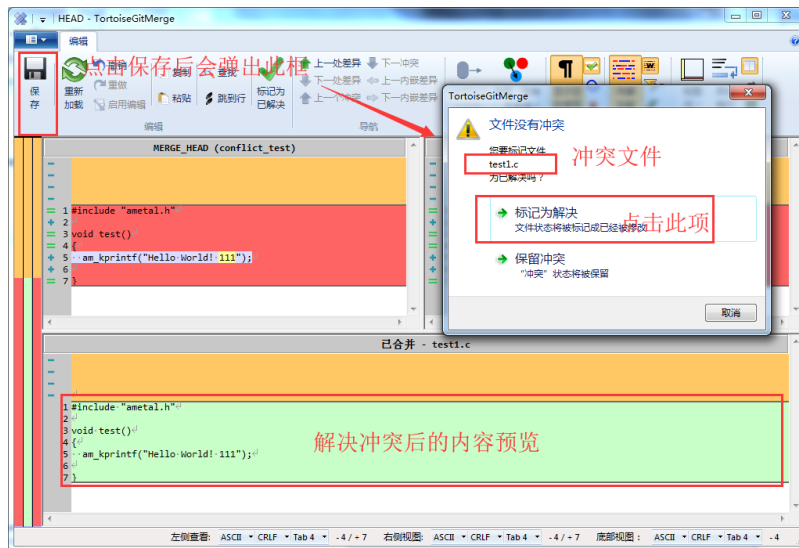


图 4.45: TortoiseGit 完成冲突解决

解决冲突时，根据实际需要选择保存内容，在图 4.44 的框 1 或者框 2 中点击右键可选择合并后使用的代码，也可在框 3 中手动输入代码进行消除冲突，最终会以框 3 内的内容为准，如图 4.45 所示，点击保存后，会弹出一个文件没有冲突提示，点击“标记为解决”项即可完成冲突解决。

但解决冲突后的代码仍是处于未提交状态，所以还需要进行代码提交，在工作目录空白处右键点击“TortoiseGit -> 提交”，会出现如图 4.46 所示界面，点击确定，提交即可。

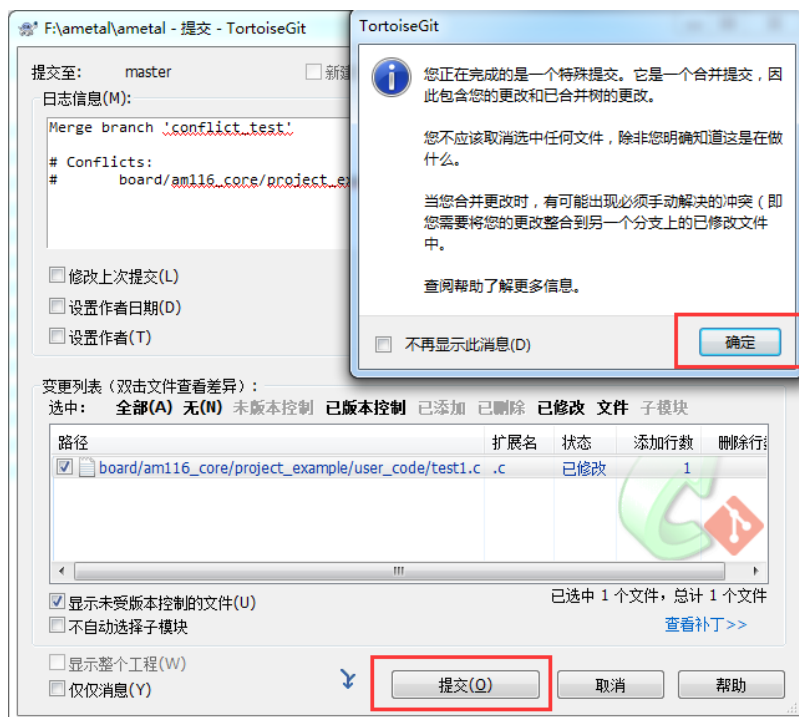


图 4.46: TortoiseGit 提交冲突解决

4.8 查看日志信息及回退版本

在版本库中，可以查看历史修改日志信息，并可以使版本库退回至指定的节点。

在本地工作目录下，在空白处单击右键，选择“TortoiseGit -> 显示日志”，会弹出一个日志信息框，如图 4.47 所示，框内有当前工作版本库的历史修改日志，包括修改时间、修改者、修改内容等信息。

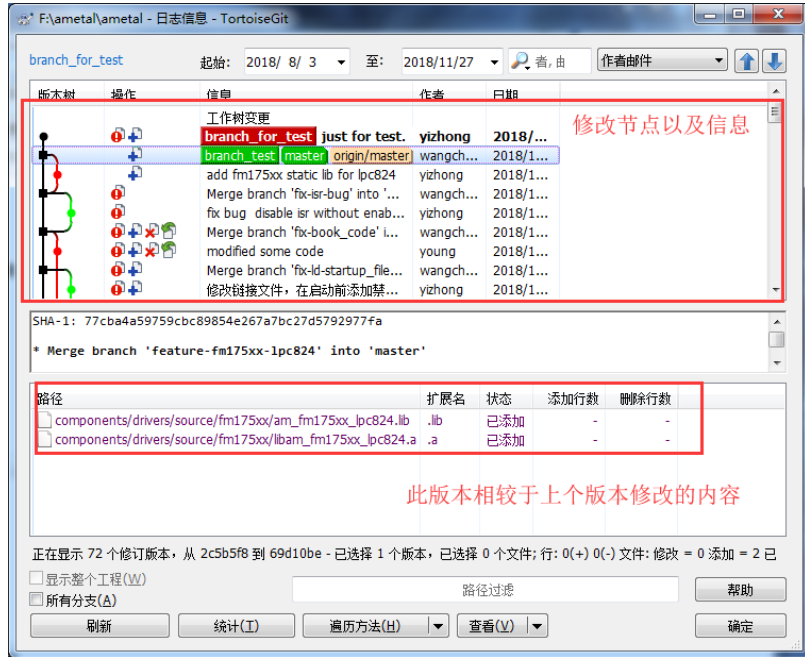


图 4.47: TortoiseGit 日志查看界面

若想跳转至某个版本库节点，则可在图 4.47 中选择相对应的节点，右键点击“重置“branch_name”到这个版本”，如图 4.48 所示，即可进行跳转至该节点操作。在弹出的重置框中直接点击“确定”按钮即可跳转至相应的节点。

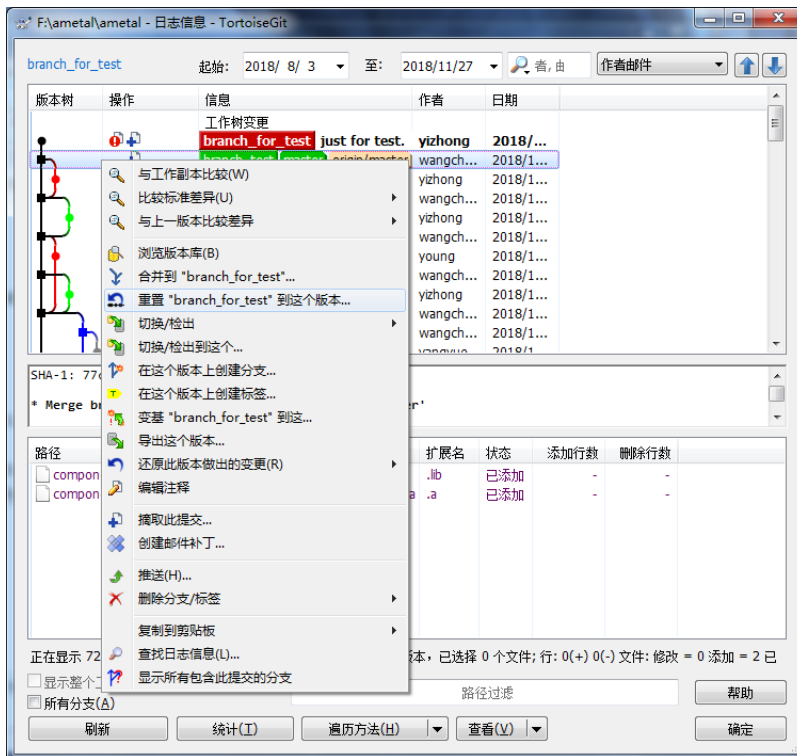


图 4.48: TortoiseGit 日志节点跳转操作

4.9 推送至远端服务器

由于权限问题，用户不能直接按照默认设置进行推送，需要将本地版本库推送至用户远端版本库，再在用户远端版本库发起“Pull Request”请求至官方版本库。

右键点击空白工作区域，选择“TortoiseGit -> 推送”项，会弹出如图 4.49 所示界面，在目标选择框中，选择“其他 URL”，并在文本框内填入用户远端版本库的 URL，点击“确定”，即可开始进行推送。

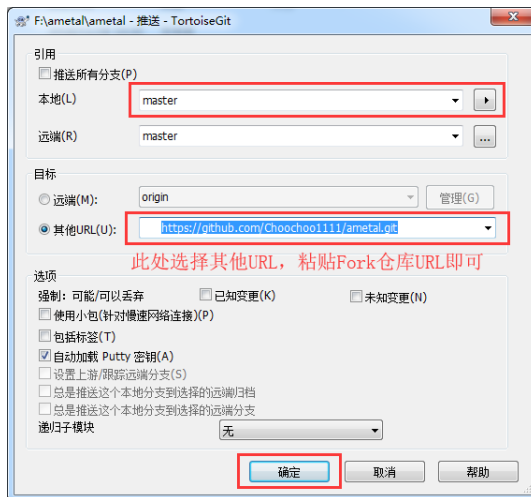


图 4.49: TortoiseGit 推送界面

若是首次进行推送，则会弹出一个 Github 登录页面，如图 4.50 所示，填写 Github 账号密码，点击“Login”，就可继续进行推送。

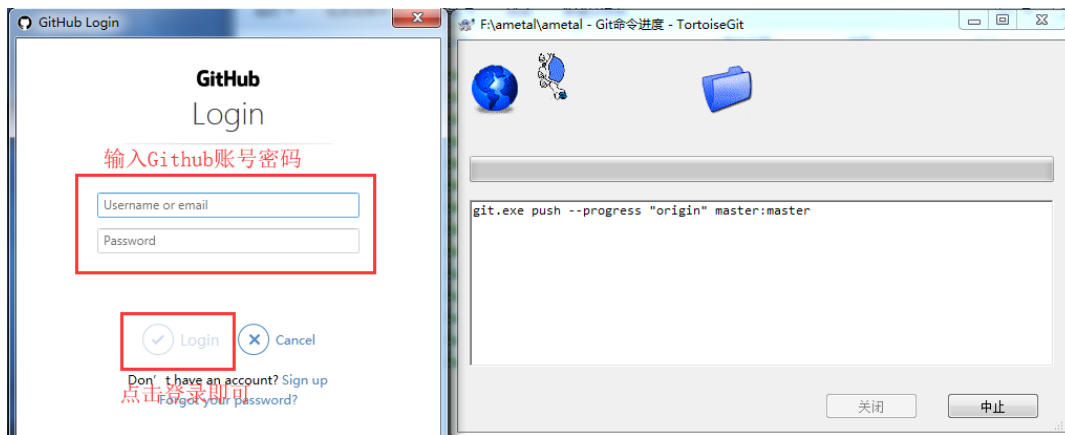


图 4.50: TortoiseGit 推送登录界面

推送完成后会弹出一个 Git 命令进度窗口，会显示推送进度，推送完成后会显示如图 4.51 所示界面，点击“关闭”即可。

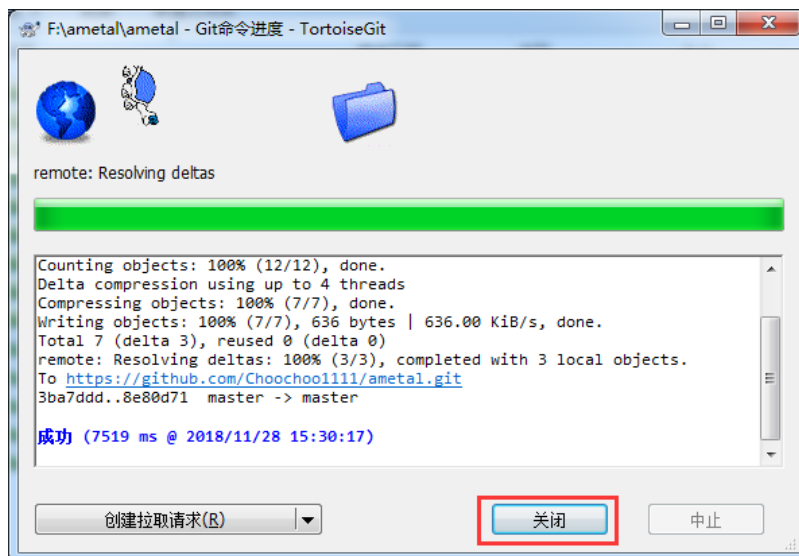


图 4.51: TortoiseGit 推送完成界面

4.10 发起“Pull Request”请求

Pull Request 是开发者使用 Github 进行协作的利器，这个功能在代码被修改并入官方版本库之前，可以得到充分的讨论，会告知官方版本库的所有参与者，可以进行代码审查，并将代码并入官方仓库的主分支。所以 Pull Request 不仅仅是一个通知，更是一个专注于某个功能的讨论版面。

推送至远端服务器之后，可以打开 Github 页面，可在账户仓库中看到版本库的内容为最新推送的内容，如图 4.52 所示。

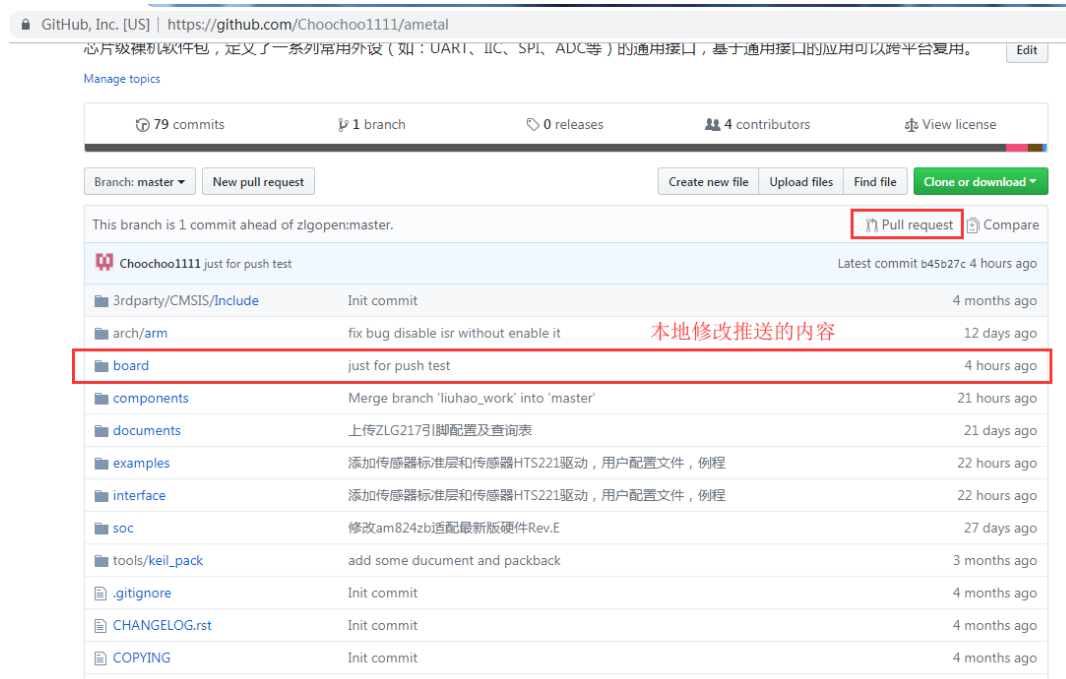


图 4.52: Github 账号用户远端版本库

若要将用户远端版本库合并至官方版本库（即 zlgopen/ametal）则需要发起“Pull Request”请求，点击图 4.52 中的“pull request”按钮，会进入如图 4.53 所示页面。在此页面可比较用户仓库与远端官方仓库不同的内容以及查看一些相关信息，并发起“Pull Request”请求。

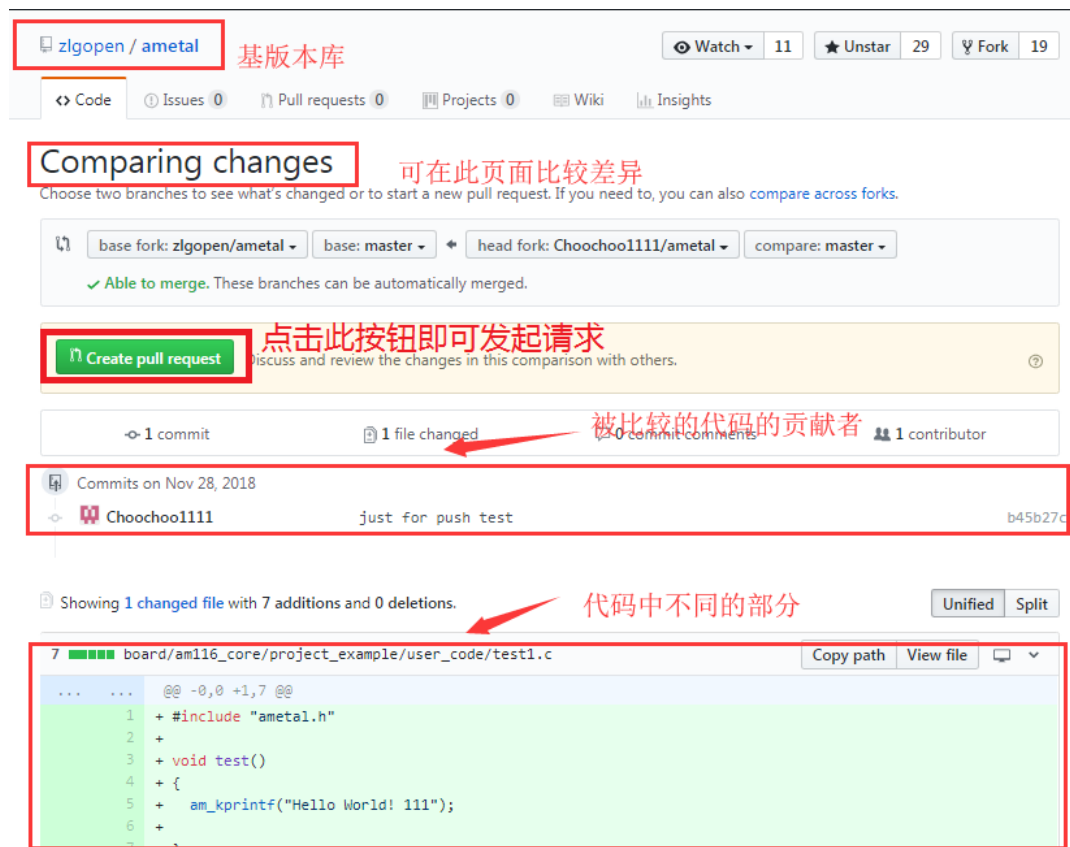


图 4.53: Github 中 Pull Request 界面

点击图 4.53 中的“Create pull request”即可发起请求，点击后会进入如图 4.54 所示界面，填写本次请求的主题以及请求的内容，然后点击按钮完成请求发起，发起完成后会进入如图 4.55 所示页面。

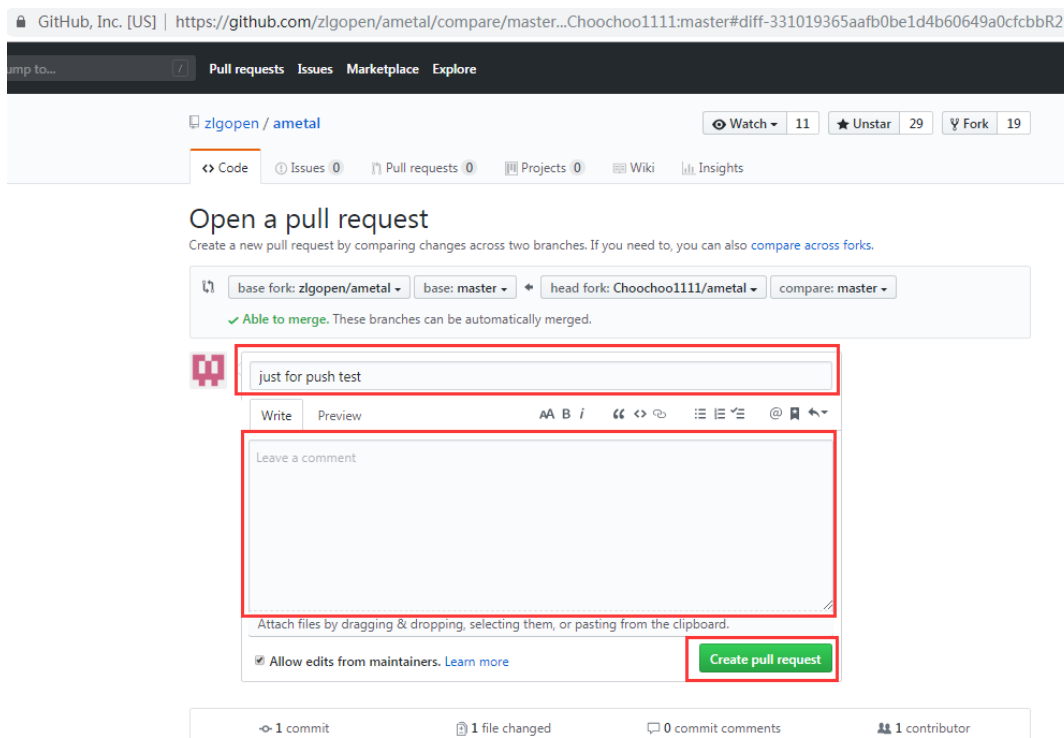


图 4.54: Github 中发起 Pull Request 请求

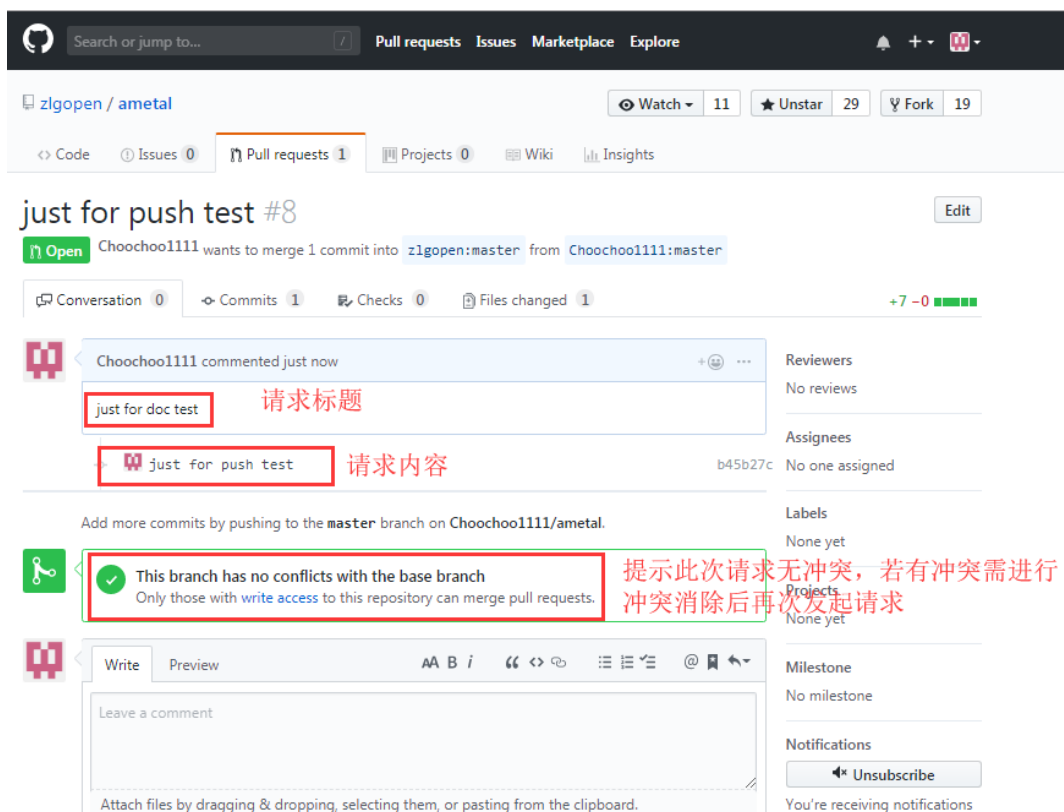


图 4.55: Github 中 Pull Request 请求发起完成页面

特别地，页面中提示了此次合并是否存在冲突，若存在冲突，用户需要消除冲突后重新进行提交、推送等一系列操作后在发起请求。

注解：请求发起后，官方版本库的管理员们将会进用户的代码进行讨论审核，若审核成功后，会同意该请求，用户的代码即会被添加至官方版本库中。

5. F&Q

5.1 创建.ssh 文件夹时，创建失败

解决：部分 Win10 不支持直接创建以“.”开头文件及文件夹，在创建文件夹时，在名字后再多加一个“.”，即文件夹名为“.ssh.”，然后就可进行创建。

5.2 在克隆以及提交部分，TortoiseGit 选项卡内无克隆及提交选项

解决：部分 Win10 克隆及提交选项位于右键菜单内。

销售与服务网络

广州周立功单片机科技有限公司

地址：广州市天河区龙怡路 117 号银汇大厦 16 楼
邮编：510630
电话：020-38730916 38730917 38730976 38730977
网址：www.zlgmcu.com
传真：020-38730925



广州专卖店

地址：广州市天河区新赛格电子城 203-204 室
电话：020-87578634/87569917
传真：020-87578842

南京周立功

地址：南京市秦淮区汉中路 27 号友谊广场 17 层 F、G 区
电话：025-68123901/68123902/68123919
传真：025-68123900

北京周立功

地址：北京市海淀区紫金数码园 3 号楼（东华合创大厦）8 层 0802 室
电话：010-62635033/62635573/62635884
传真：010-82164433

重庆周立功

地址：重庆市渝北区龙溪街道新溉大道 18 号山顶国宾城 11 幢 4-14
电话：023-68796438/68796439/68797619
传真：023-68796439

杭州周立功

地址：杭州市西湖区紫荆花路 2 号杭州联合大厦 A 座 4 单元 508
电话：0571-89719484/89719499/89719498
传真：0571-89719494

成都周立功

地址：成都市一环路南二段 1 号数码科技大厦 403 室
电话：028-85439836/85432683/85437446
传真：028-68796439

深圳周立功（一部）

地址：深圳市福田区深南中路 2072 号电子大厦 1203 室
电话：0755-82941683/82907445
传真：0755-83793285

深圳周立功（二部）

地址：深圳市坪山区比亚迪路大万文化广场 A 座 1705
电话：0755-83781788/83782922
传真：0755-83793285

武汉周立功

地址：武汉市武昌区武珞路 282 号思特大厦 807 室
电话：027-87168497/87168297/87168397
传真：027-87163755

上海周立功

地址：上海市黄浦区北京东路 668 号科技京城东座 12E 室
电话：021-53083451/53083452/53083453
传真：021-53083491

周立功厦门办

地址：厦门市思明区厦禾路 855 号英才商厦 618 室
电话：18650195588

周立功苏州办

地址：江苏省苏州市广济南路 258 号（百脑汇科技中心 1301 室）
电话：0512-68266786 & 18616749830

周立功合肥办

地址：安徽省合肥市蜀山区黄山路 665 号汇峰大厦 1607
电话：13851513746

周立功宁波办

地址：浙江省宁波市高新区星海南路 16 号轿辰大厦 1003
电话：0574-87228513/87229313

周立功天津办

地址：天津市河东区十一经路与津塘公路交口鼎泰大厦 1004 室
电话：18622359231

周立功山东办

地址：山东省青岛市李沧区青山路 689 号宝龙公寓 3 号楼 311
电话：13810794370

周立功郑州办

地址：河南郑州市中原区百花路与建设路东南角锦绣华庭 A 座 1502 室
电话：17737307206

周立功沈阳办

地址：沈阳市浑南新区营盘西街 17 号万达广场 A4 座 2722 室
电话：18940293816

香港周立功

地址：香港新界沙田火炭禾香街 9-15 力坚工业大厦 13 层
电话：(852)26568073 26568077

周立功长沙办

地址：湖南省长沙市岳麓区奥克斯广场国际公寓 A 栋 2309 房
电话：0731-85161853