HelloX项目协同开发指南

## 概述

为了提高协同开发效率，HelloX项目已托管到github网站上。根据目前的开发进展，创建了下列几个子项目：

HelloX操作系统内核项目：<https://github.com/hellox-project/HelloX_Kernel.git>

HelloX操作系统GUI模块项目：<https://github.com/hellox-project/HelloX_GUI.git>

面向STM32的移植版：<https://github.com/hellox-project/HelloX_STM32.git>

后续根据需要，再增加其它相关子项目，所有这些子项目都是HelloX项目的一部分。

本文档对如何协同开发进行描述，旨在帮助开源开发人员顺利搭建起github协作开发环境，共同开发HelloX项目。

## Github工作原理简介

严格来说，github是一个网站的名称，真正的版本管理功能是由git实现的。为了简便，笼统的认为github就是一个版本管理系统就OK了。

其实，github的工作原理非常简单，借用我们日常开发时使用的策略类比，会很容易理解。比如，现在有一个开发项目，这个开发项目经过长时间积累，已经有了数万行代码，并由一个组织或个人唯一管理（如果多个人同时管理，就乱套了）。你希望对这个项目进行修改或者优化。这时候，谨慎起见，你不可能直接在原项目上进行修改，而是把原来的项目复制一份，在这个复制项目上进行修改。修改完毕后，会总结出修改了哪些内容（修改了哪些文件，添加了哪些文件，删除了哪些文件等），然后把修改的内容提交给管理者审核，审核通过后，管理者会逐一合并到原有项目中。

Github自动化的实现了这个过程。原始项目存放在一个固定的位置，并由一个作者同意管理。比如对于HelloX项目的HelloX操作系统内核源代码，其固定存放位置是github.com/hellox-project/HelloX\_Kernel，当前的管理者是我本人。如果你希望参与开发，对内核进行修改，那么就需要首先复制一份（github上叫做fork），到你自己的位于github网站的个人空间内。这时候你就可以随便修改这份copy了。但由于这份拷贝是在网上，无法直接修改，因此你必须把这份copy下载到本地（github叫做clone），然后在本地进行修改。Github（严格来说，是github的本地工具）会监控这个修改过程，把主要的修改内容都记录下来（之所以记录过程，是方便倒回）。本地修改完之后，你告诉github（叫做commit），已经修改完成了，github会把修改的内容统一合并到本地代码库中。

注意，这时候你的修改还是停留在本地代码库中，尚未传递到你自己的网上空间内。如果要把修改内容上传到你自己的github空间，需要执行一个叫做push的操作，把本地commit的修改，推送到你自己的github空间内。

一旦push完成，修改才同步到你自己的github空间中。为了把你的修改合并到原始代码中，你必须发起一个pull request，告诉项目管理者，你做了哪些修改，为什么做这些修改，申请项目管理者合并到主流版本中。剩下的事情，就是项目管理者的事了，他会考虑这些修改申请，必要时会发起讨论，或者跟你联系，讨论这些修改的必要性。如果最终确定这些修改是有价值的，那么项目管理者会merge到主流版本中，并记录你的contribution。

好了，通过上面的描述，记住并理解下面的关键词，后面的操作就非常简单了：

Fork，clone，commit，push，merge

## HelloX项目github协作开发指南

首先，你必须了解HelloX项目的背景及目标，判断是否与您的背景或者意图能够匹配，以决定是否要投入开发。这个过程要慎重仔细，一旦决定要投入，那么就需要遵循开源精神，有始有终，当然，您的贡献会被尽量精确的记录下来（HelloX项目的每个子项目下，都有authors.txt文件，记录了贡献者的主要贡献及个人信息）。

在决定投入开发之后，剩下的事情就很简单了。下面一步一步的介绍。

### 第一步：在github上注册

在github上注册一下，创建自己的user space。这个过程不细说，直接登录github.com，点击sign up即可。建议选择注册的用户名、邮箱、密码等信息，万一您成为github上的大牛，一个很酷的名字会很重要。

### 第二步：fork待参与的项目

在github上注册成功之后，会一直保持登录。这时候，建议fork一下您希望参与开发的项目，比如您希望参与HelloX项目中的HelloX操作系统STM32版子项目，那么在浏览器地址栏中输入<https://github.com/hellox-project>，即可连接到hellox项目的主页下，页面上会列出所有子项目。选择其中的一个，进去之后，点击页面上的fork按钮，即可复制一份到您自己的空间。

再回到您自己的个人页面，即可看到fork过来的项目。

### 第三步：下载并安装github客户端

下列地址可以下载github客户端：

<https://help.github.com/articles/set-up-git>

下载后，直接运行即可。注意，根据我的经验，上面网页上下载的只是一个launch程序，并不是真正的github安装程序。这个launch程序会进一步从github网站上（貌似托管在Amazon的云上）下载安装程序。这个过程会比较长，网络情况不好的话，可能会失败。请多试几次。

在安装过程中，会提示您输入第一步创建的github账户信息。

### 第四步：启动github命令行客户端，下载项目

最新的github版本，提供图形界面和命令行界面。虽然图形界面做得很简洁很好看，但用起来还是感觉不太方便，我个人仍然倾向于github命令行客户端。

按照缺省安装，github会在桌面上生成一个git shell的图标，双击即可进入命令行界面。这个命令行界面整合了github的相关命令和windows相关命令，比原来的unix命令行风格好用了很多。

创建一个本地工作目录，比如D:\HelloX，然后使用CD等命令，定位到工作目录，运行下列命令：

D:\HelloX>git clone https://github.com/*yourname*/HelloX\_STM32.git

即可把您在github上fork的项目，下载到本地。注意把上述URL中的用户名，换成你自己的github用户名。

下载完毕之后，进入新创建的项目目录（位于您的工作目录下），然后会发现命令行提示符变了。运行git status命令，即可看到当前项目的状态。

### 第五步：对源代码进行编辑和修改

这一步是核心工作，使用编译器打开下载的项目，比如用MDK，可以直接打开STM32版的HelloX内核项目。遵循正常的程序编写流程，对源代码进行修改即可。

### 第六步：提交本地修改

第五步修改完成之后，您所做的修改只是停留在源代码文件中，尚未被github接纳到它自己的数据库中。这时候，必须在命令行下执行commit操作。过程如下：

1. 执行git status命令，会列做了修改的文件，包括增加的源代码文件、删除的源代码文件；
2. 执行git add . 命令（注意命令后面的点号），把修改增加到github管理的数据库中；
3. 执行git commit –m “修改描述” 命令，把修改提交到github本地库中。注意，引号中的描述，是对本次修改的简要介绍，会呈现在源代码文件的修改历史中。

完成上述步骤之后，您所做的修改就记录到github本地库中了。

如果做进一步的源代码修改，修改完成后，重复上述操作，提交到本地库中。

### 第七步：把本地修改上载到github

在本地代码库上完成提交之后，修改还是局限在本地库，尚未上传到github的个人空间内。这时候执行下列命令，把修改的代码同步到您自己的github空间中：

D:\HelloX>git push –u origin master

其中的origin，代表代码的原始版本，即是您自己在github上fork的版本（注意不是项目的最原始版本）。Master则是本地版本。

### 第八步：发起pull request，请求合并修改

Push到个人github空间之后，如果希望合并到项目的主版本中，必须发起pull request，让项目管理员来审核您所做的修改，并决定是否合并到项目主版本中。具体操作是，在您自己的github项目页面上，点击pull request，会显示出创建pull request的页面。如果感觉页面内容很复杂，不要紧，直接点击create pull request即可，大部分内容，github已帮助您填好了。

到此，您的大部分工作就完成了，接下来就是项目管理员审核pull request，并讨论是否最终合并到版本中。如果确定要合并，那么您的contribution会被记录下来。