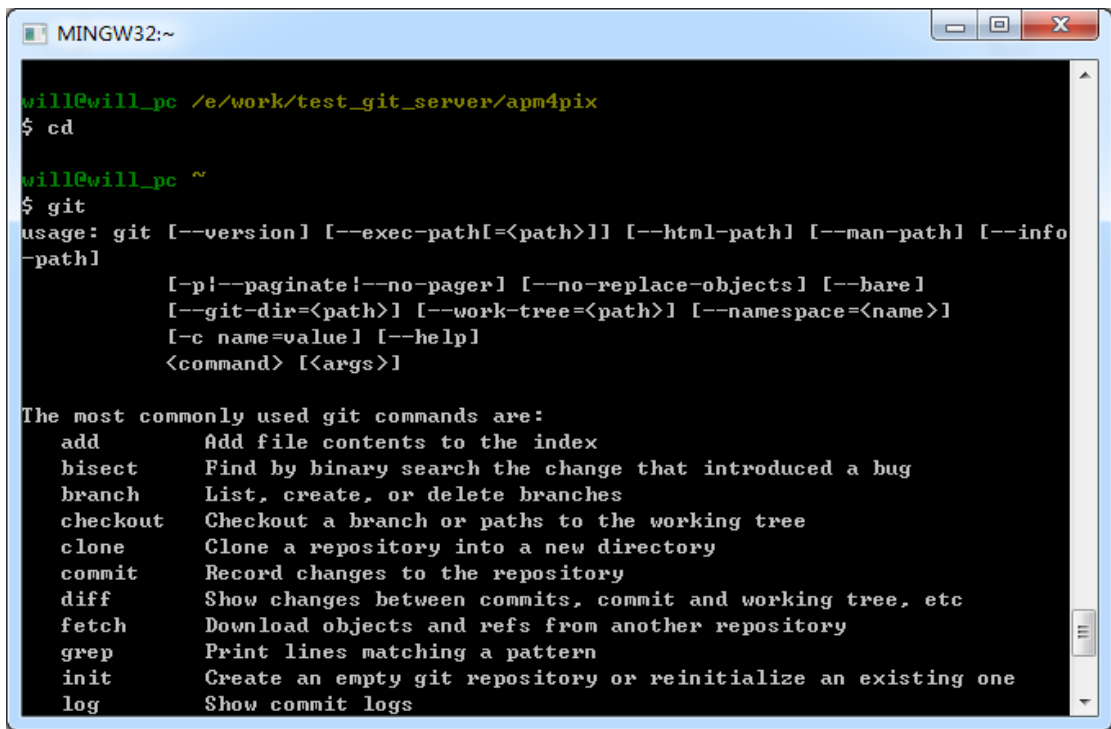


局域网内采用 git 版本控制的应用

一. 安装 git 客户端, <https://git-scm.com/>

安装完成后, 在 px4 console 中输入 **git**, 提示 git usage: , 表示安装成功。如图:



```
will@will_pc /e/work/test_git_server/apm4pix
$ cd

will@will_pc ~
$ git
usage: git [--version] [--exec-path[=<path>]] [--html-path] [--man-path] [--info-
-path]

        [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        [-c name=value] [--help]
        <command> [<args>]

The most commonly used git commands are:
  add          Add file contents to the index
  bisect       Find by binary search the change that introduced a bug
  branch       List, create, or delete branches
  checkout     Checkout a branch or paths to the working tree
  clone        Clone a repository into a new directory
  commit       Record changes to the repository
  diff         Show changes between commits, commit and working tree, etc
  fetch        Download objects and refs from another repository
  grep         Print lines matching a pattern
  init         Create an empty git repository or reinitialize an existing one
  log          Show commit logs
```

二. 配置 git, 配置 user_name 和 user_email, 需要填入自己的用户名和邮箱

git config --global user.name "user_name" 回车确认

git config --global user.email "user_email@xx.com" 回车确认

ssh-keygen -t rsa -C "user_email@xx.com" 回车确认

一直回车确认, 就可以生成 id_rsa 和 id_rsa.pub, 其存放在

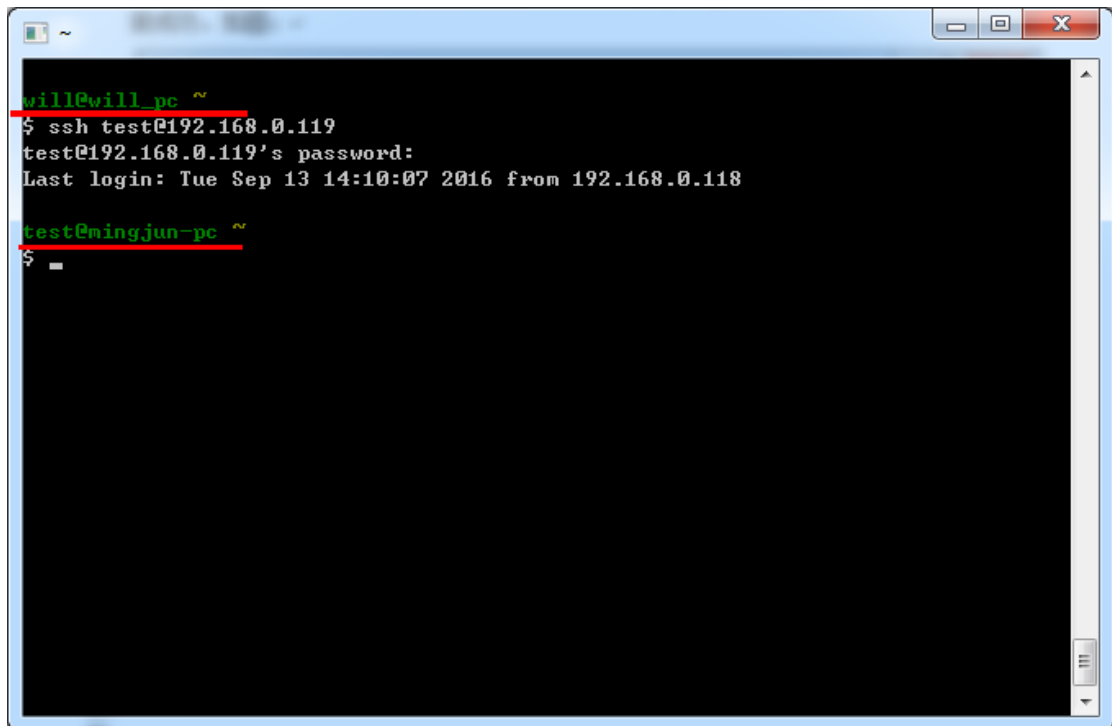
d:\px4\toolchain\msys\1.0\home\will\.ssh 下

三. 登录 ssh 服务器, 密码为 test , 如图。

首次登录会确认是否添加到信任, 输入 **yes**

ssh test@192.168.0.119

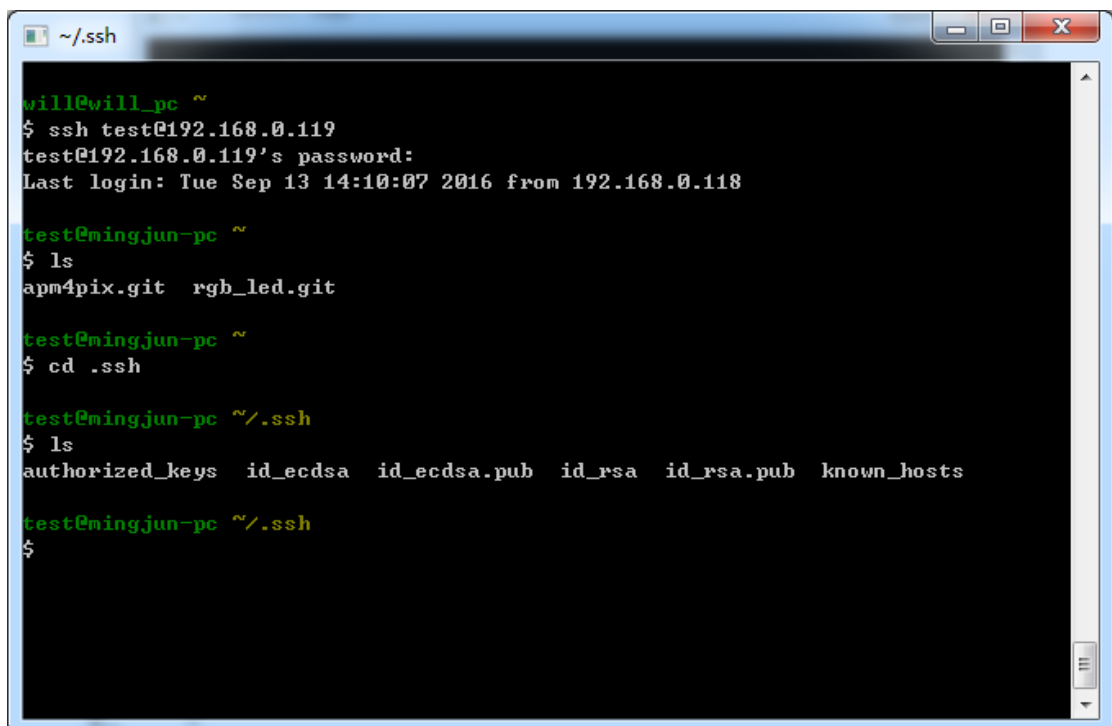
密码为 **test**



```
will@will_pc ~  
$ ssh test@192.168.0.119  
test@192.168.0.119's password:  
Last login: Tue Sep 13 14:10:07 2016 from 192.168.0.118  
test@mingjun-pc ~  
$
```

四. 进入到服务器端存放 ssh_key 的目录下

`cd .ssh` 回车确认, 此目录为隐藏

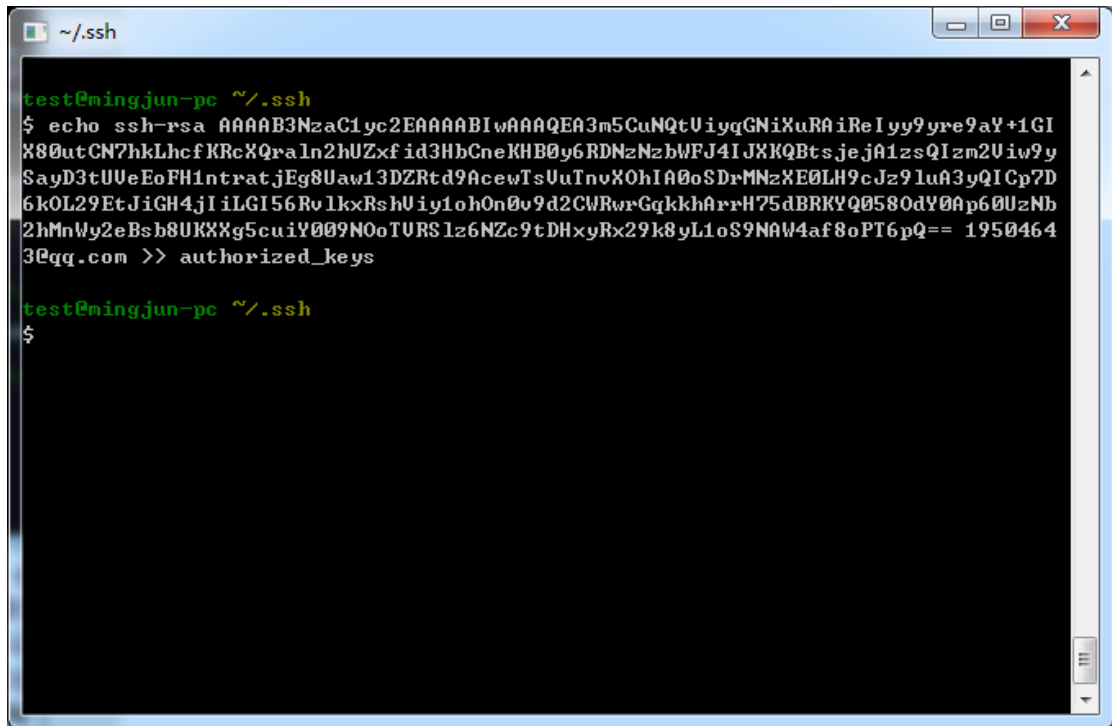


```
~/ssh  
will@will_pc ~  
$ ssh test@192.168.0.119  
test@192.168.0.119's password:  
Last login: Tue Sep 13 14:10:07 2016 from 192.168.0.118  
test@mingjun-pc ~  
$ ls  
apm4pix.git  rgb_led.git  
test@mingjun-pc ~  
$ cd .ssh  
test@mingjun-pc ~/.ssh  
$ ls  
authorized_keys  id_ecdsa  id_ecdsa.pub  id_rsa  id_rsa.pub  known_hosts  
test@mingjun-pc ~/.ssh  
$
```

五. 将本地生成的 id_rsa.pub 添加到 authorized_keys 中, 以后登录服务器、git clone、git pull、git push 等操作就都不需要密码, 而采用 ssh 协议加密, 更加安全, 步骤如下:

1. 用 notepad++ 或记事本打开本地 `d:\px4\toolchain\msys\1.0\home\will\.ssh` 下的 `id_rsa.pub`, 删除第二行, 只剩第一行, 然后全选复制。

2. 到 px4 console, 输入 `echo` 后, 右键粘贴, 然后输入 `>> authorized_keys`, 如图



```
test@mingjun-pc ~/.ssh
$ echo ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA3m5CuNQtUiyyqGNiXuRAiReIyy9yre9aY+1GI
X80utCN7hkLhcfKRcXQraIn2hUZxfid3HbCneKHB0y6RDNzNzbWfJ4IJXKQBtsjeJA1zsQIzm2Uiw9y
SayD3tUUEoFH1ntrajtEg8Uaw13DZRtd9AcawTsUuTnuXOhIA0o$DrMNzXE0LH9cJz9luA3yQICp7D
6kOL29EtJiGH4jIiLGI56Rv1kxRshUiy1oHOn0v9d2CWRwrGqkkhArrH75dBRKYQ0580dY0Ap60UzNb
2hMnWy2eBsb8UKXg5cuiY009N0oTURS1z6NZc9tDHxyRx29k8yL1oS9NAW4af8oPT6pQ== 1950464
3Eqq.com >> authorized_keys

test@mingjun-pc ~/.ssh
$
```

- 六. 测试 ssh 是否认证成功, 要先退出登录, 重新登录服务器, 输入

`exit` 回车确认, 退出当前登录, 回到自己的客户机, 然后输入

`ssh test@192.168.0.119` 回车确认

此时就可以不需要输入密码登录, 再次退出登录

`exit`

- 七. 从服务器端 clone 代码仓库, 目前服务器端已有两个仓库, 分别为 `apm4pix.git` 和 `rgb_led.git`, 如 clone 第一个仓库到 E 盘 `work` 目录下, 如图

`mkdir /f/work && cd /f/work`

`git clone test@192.168.0.119:apm4pix`

```
MINGW32:/f/work

will@will_pc ~
$ mkdir /f/work && cd /f/work

will@will_pc /f/work
$ git clone test@192.168.0.119:apm4pix
Cloning into 'apm4pix'...
remote: Counting objects: 125400, done.
remote: Compressing objects: 100% (31238/31238), done.
remote: Total 125400 (delta 92034), reused 125285 (delta 91967)
Receiving objects: 100% (125400/125400), 91.44 MiB | 1.06 MiB/s, done.

Resolving deltas: 100% (92034/92034), done.

will@will_pc /f/work
$
```

八. 进入 apm4pix 目录，就可以编译 pixhawk 固件了

```
cd apm4pix/arducopter
make px4-v2
```

```
MINGW32:/f/work/apm4pix/arducopter

remote: Compressing objects: 100% (31238/31238), done.
remote: Total 125400 (delta 92034), reused 125285 (delta 91967)
Receiving objects: 100% (125400/125400), 91.44 MiB | 1.06 MiB/s, done.

Resolving deltas: 100% (92034/92034), done.

will@will_pc /f/work
$ cd apm4pix/arducopter

will@will_pc /f/work/apm4pix/arducopter
$ make px4-v2
// BUILDROOT=/f/work/apm4pix/Build.arducopter HAL_BOARD=HAL_BOARD_PX4 HAL_BOARD_
SUBTYPE= TOOLCHAIN=NATIVE EXTRAFLAGS=-DGIT_VERSION="b1966fc5" -DNUTTX_GIT_VERSION="b1966fc5" -DPX4_GIT_VERSION="b1966fc5"
Checking modules
make[1]: Entering directory `/f/work/apm4pix/modules/PX4Firmware'
Skipping submodules. NUTTX_SRC is set to /f/work/apm4pix/modules/PX4NuttX/nuttx/

%% Configuring NuttX for px4fmu-v2
make[3]: Nothing to be done for `clean'.
make[5]: Nothing to be done for `clean'.
make[5]: Nothing to be done for `clean'.
make[5]: Nothing to be done for `clean'.
make[5]: Nothing to be done for `clean'.
make[5]: Nothing to be done for `clean'.
```

编译成功后，如下图

```
MINGW32:/f/work/apm4pix/arducopter
ommands.c
Pruning ROMFS files.
ROMFS:  romfs.img
OBJ:    romfs.o
CC:     romfs.o.c
CC:     romfs.o.c
LINK:   /f/work/apm4pix/modules/PX4Firmware/Build/px4fmu-v2_APM.build/firmware.elf
BIN:    /f/work/apm4pix/modules/PX4Firmware/Build/px4fmu-v2_APM.build/firmware.bin
%% Generating /f/work/apm4pix/modules/PX4Firmware/Build/px4fmu-v2_APM.build/firmware.px4
fatal: Not a git repository: 'f:/work/apm4pix/modules/PX4Firmware/..git'
make[12]: Leaving directory '/f/work/apm4pix/modules/PX4Firmware/Build/px4fmu-v2_APM.build'
%% Copying /f/work/apm4pix/modules/PX4Firmware/Images/px4fmu-v2_APM.px4
make[11]: Leaving directory '/f/work/apm4pix'
fatal: Not a git repository: 'f:/work/apm4pix/modules/PX4Firmware/.git'
Failed to get px4 hash
fatal: Not a git repository: 'f:/work/apm4pix/modules/PX4NuttX/nuttX/../../.git'
Failed to get nuttx hash
PX4 arducopter Firmware is in arducopter-v2.px4
will@will_pc /f/work/apm4pix/arducopter
$
```

九. 对固件进行版本控制

输入 **git log**, 就可以查看更改历史, 具体其他方法, 请参看 git 文档

```
MINGW32:/f/work/apm4pix/arducopter
commit b1966fc527b7b59b26cce25c22030730ee92518f
Author: 19504643 <19504643@qq.com>
Date:   Wed Sep 7 17:56:31 2016 +0800

    solved safety switch bug when arming

commit e7c0d5397235c30f771c44430e7d7cbb27d23387
Author: 19504643 <19504643@qq.com>
Date:   Wed Sep 7 15:51:37 2016 +0800

    add canlink protocol success

commit 4b390dd30610d9d640f0b46581c739de0b17561f
Author: 19504643 <19504643@qq.com>
Date:   Wed Aug 24 09:44:30 2016 +0800

    can transceiver works,ready to add protocol

commit abeff750db0f72d449e31d29b738c7a876839789
Author: 19504643 <19504643@qq.com>
Date:   Mon Aug 22 14:32:58 2016 +0800

    add track of ../modules/PX4Firmware/src/modules/uORB/topics/
:
```

此时输入 **git status**, 会提示有很多文件修改过

```
will@will_pc /f/work/apm4pix/arducopter
$ git status
# On branch master
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   ../module.mk
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_armed.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_controls.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_controls_0.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_controls_1.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_controls_2.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_controls_3.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_control_virtual_fw.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_control_virtual_mc.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/actuator_directives.h
```

这是因为在提交到服务器后，再 clone 到本地后，可能路径发生了变化
用 `git diff` 查看

```
git diff ../modules/PX4Firmware/src/modules/uORB/topics/wind estimate.h
```

```

MINGW32:/f/work/apm4pix/ArduCopter
no changes added to commit (use "git add" and/or "git commit -a")

will@will_pc /f/work/apm4pix/ArduCopter
$ git diff ../modified: ../m
fatal: Invalid object name 'modified'.

will@will_pc /f/work/apm4pix/ArduCopter
$ git diff ../modules/PX4Firmware/src/modules/uORB/topics/wind_estimate.h
diff --git a/modules/PX4Firmware/src/modules/uORB/topics/wind_estimate.h b/modul
index bd48661..78d5649 100644
--- a/modules/PX4Firmware/src/modules/uORB/topics/wind_estimate.h
+++ b/modules/PX4Firmware/src/modules/uORB/topics/wind_estimate.h
@@ -31,7 +31,7 @@
 *
 *****/

-/* Auto-generated by genmsg_cpp from file e:\OpenSource\ardupilot\modules\PX4Fi
+/* Auto-generated by genmsg_cpp from file f:\work\apm4pix\modules\PX4Firmware\m

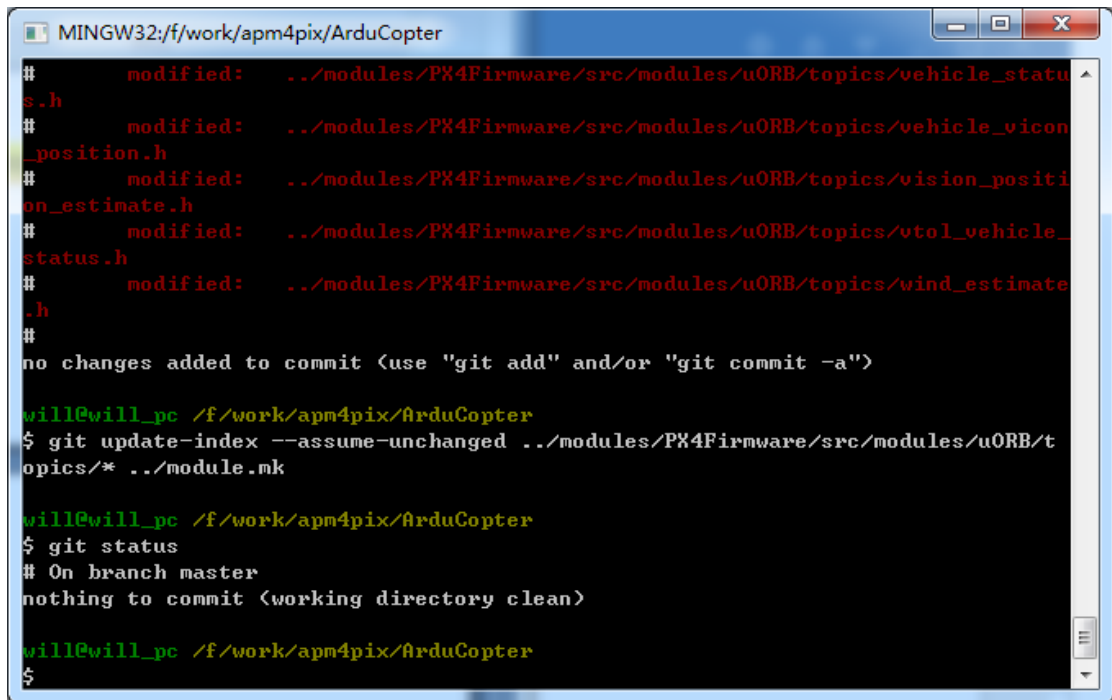
#pragma once
warning: CRLF will be replaced by LF in modules/PX4Firmware/src/modules/uORB/top
The file will have its original line endings in your working directory.
<END>

```

可以用如下命令，取消跟踪

```
git update-index
```

```
--assume-unchanged ../modules/PX4Firmware/src/modules/uORB/topics/* ../modules/
module.mk
```



```
MINGW32:/f/work/apm4pix/ArduCopter
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/vehicle_status.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/vehicle_vicon_position.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/vision_position_estimate.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/vtol_vehicle_status.h
#       modified:   ../modules/PX4Firmware/src/modules/uORB/topics/wind_estimate.h
#
no changes added to commit (use "git add" and/or "git commit -a")

will@will_pc /f/work/apm4pix/ArduCopter
$ git update-index --assume-unchanged ../modules/PX4Firmware/src/modules/uORB/topics/* ../module.mk

will@will_pc /f/work/apm4pix/ArduCopter
$ git status
# On branch master
nothing to commit (working directory clean)

will@will_pc /f/work/apm4pix/ArduCopter
$
```

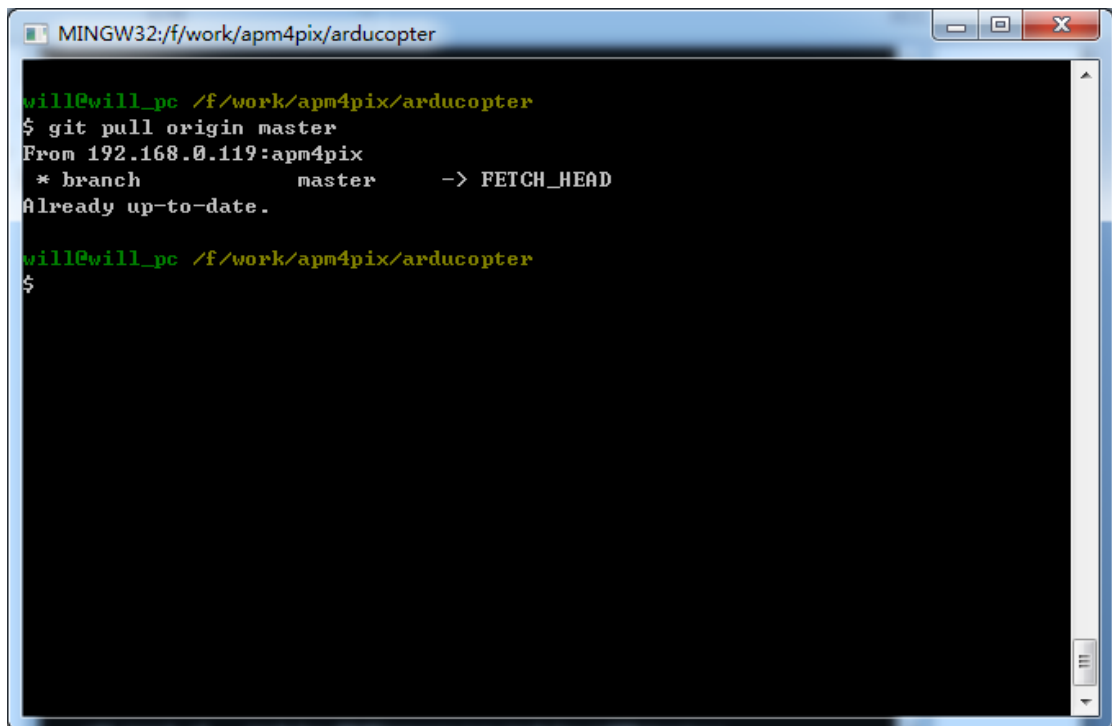
然后输入 **git status** 查看

十. 多人协作

1、每次修改各自的版本后，需要提交到服务器

提交如果失败，说明有人已经提交过，需要先 pull 服务器端的文件到本地

git pull origin master 如果当前版本为 master，如图



```
MINGW32:/f/work/apm4pix/arducopter

will@will_pc /f/work/apm4pix/arducopter
$ git pull origin master
From 192.168.0.119:apm4pix
 * branch          master      -> FETCH_HEAD
Already up-to-date.

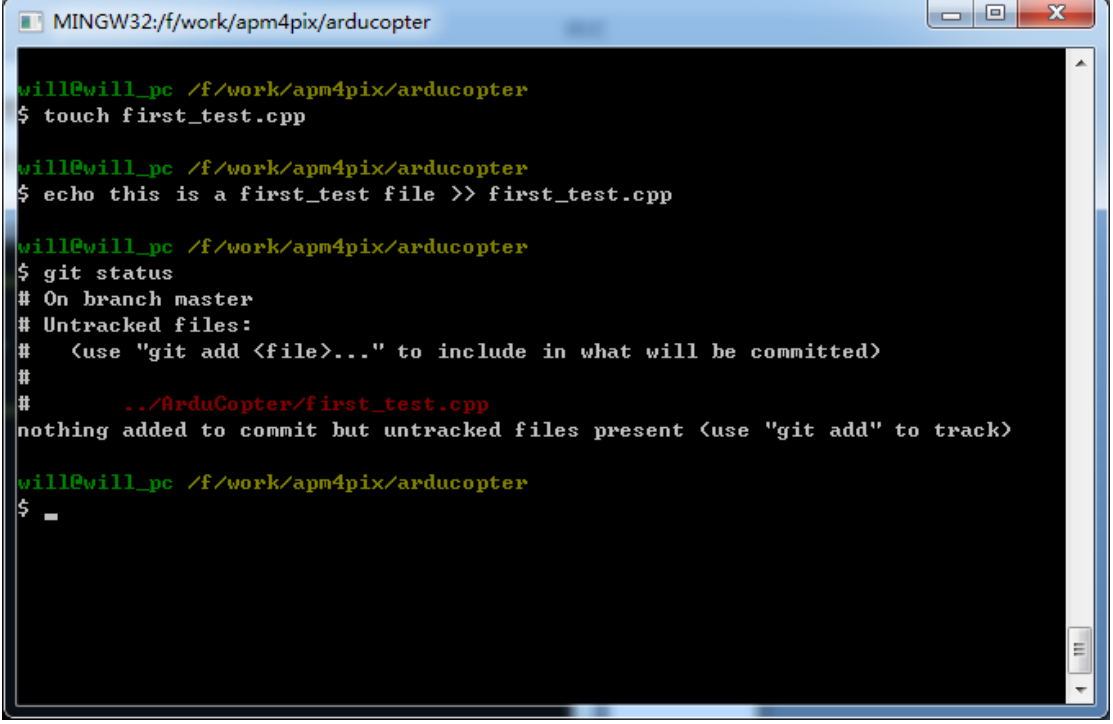
will@will_pc /f/work/apm4pix/arducopter
$
```

如果服务器端文件比本地文件新，需要 merge，这个参看 [git 手册](#)

2、在 arducopter 文件夹下新建一个文件 first_test.cpp

```
touch first_test.cpp
```

```
echo this is a first_test file >>first_test.cpp
```

A screenshot of a terminal window titled 'MINGW32:/f/work/apm4pix/arducopter'. The terminal shows a series of commands and their outputs. The user runs 'touch first_test.cpp', then 'echo this is a first_test file >> first_test.cpp', and finally 'git status'. The output of 'git status' indicates that the file is untracked and not yet added to the commit.

```
MINGW32:/f/work/apm4pix/arducopter

will@will_pc /f/work/apm4pix/arducopter
$ touch first_test.cpp

will@will_pc /f/work/apm4pix/arducopter
$ echo this is a first_test file >> first_test.cpp

will@will_pc /f/work/apm4pix/arducopter
$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       ../ArduCopter/first_test.cpp
nothing added to commit but untracked files present (use "git add" to track)

will@will_pc /f/work/apm4pix/arducopter
$
```

然后输入

```
git add .
```

```
git commit -m 'create first_test.cpp'
```

```
git push origin master
```



```
MINGW32:/f/work/apm4pix/arducopter

#
#      ../ArduCopter/first_test.cpp
nothing added to commit but untracked files present (use "git add" to track)

will@will_pc /f/work/apm4pix/arducopter
$ git add .

will@will_pc /f/work/apm4pix/arducopter
$ git commit -m 'create first_test.cpp'
[master e60f4f7] create first_test.cpp
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 ArduCopter/first_test.cpp

will@will_pc /f/work/apm4pix/arducopter
$ git push origin master
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 360 bytes, done.
Total 4 (delta 2), reused 0 (delta 0)
To test@192.168.0.119:apm4pix
    b1966fc..e60f4f7  master -> master

will@will_pc /f/work/apm4pix/arducopter
$
```

此时 **git log**

```
MINGW32:/f/work/apm4pix/arducopter

commit e60f4f7da8cb32a2aaa1053b7c8858d9c69ba76d
Author: 19504643 <19504643@qq.com>
Date:   Tue Sep 13 15:24:22 2016 +0800

    create first_test.cpp

commit b1966fc527b7b59b26cce25c22030730ee92518f
Author: 19504643 <19504643@qq.com>
Date:   Wed Sep 7 17:56:31 2016 +0800

    solved safety switch bug when arming

commit e7c0d5397235c30f771c44430e7d7cbb27d23387
Author: 19504643 <19504643@qq.com>
Date:   Wed Sep 7 15:51:37 2016 +0800

    add canlink protocol success

commit 4b390dd30610d9d640f0b46581c739de0b17561f
Author: 19504643 <19504643@qq.com>
Date:   Wed Aug 24 09:44:30 2016 +0800

    can transceiver works,ready to add protocol

:
```

如何将自己的代码库上传到服务器

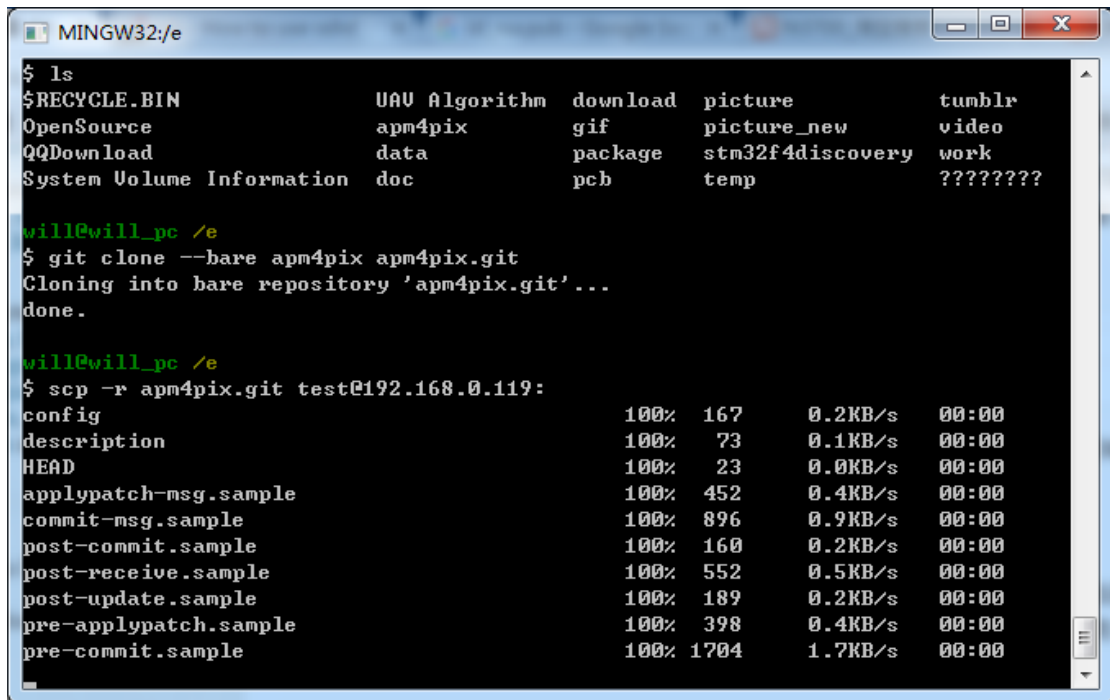
到本地代码目录的上一级目录，如代码路径为 e:/apm4pix，则

cd /e

git clone --bare apm4pix apm4pix.git

scp -r apm4pix.git test@192.168.0.119:

这样就上传完成，其他人就可以进行 clone 和版本控制



```
MINGW32/e
$ ls
$RECYCLE.BIN      UAV Algorithm  download  picture      tumblr
OpenSource        apm4pix       gif       picture_new  video
QQDownload        data         package   stm32f4discovery work
System Volume Information doc          pcb       temp         ????????

will@will_pc /e
$ git clone --bare apm4pix apm4pix.git
Cloning into bare repository 'apm4pix.git'...
done.

will@will_pc /e
$ scp -r apm4pix.git test@192.168.0.119:
config                100% 167      0.2KB/s  00:00
description           100%  73      0.1KB/s  00:00
HEAD                  100%  23      0.0KB/s  00:00
applypatch-msg.sample 100% 452      0.4KB/s  00:00
commit-msg.sample     100% 896      0.9KB/s  00:00
post-commit.sample    100% 160      0.2KB/s  00:00
post-receive.sample   100% 552      0.5KB/s  00:00
post-update.sample    100% 189      0.2KB/s  00:00
pre-applypatch.sample 100% 398      0.4KB/s  00:00
pre-commit.sample     100% 1704     1.7KB/s  00:00
```

文中黄底文字，是可以执行的命令