

一、依赖库与软件安装教程

1. 安装依赖

```
sudo apt install mesa-common-dev freeglut3-dev coinor-libipopt-dev libblas-dev liblapack-dev gfortran liblapack-dev coinor-libipopt-dev cmake gcc build-essential libglib2.0-dev
```

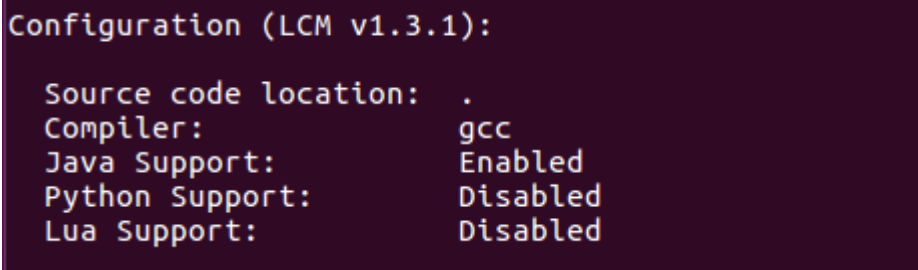
2. 安装 openjdk

```
sudo apt-get update  
sudo apt-get install openjdk-8-jdk
```

3. 安装 lcm1.3.1

下载 lcm1.3.1 包进行解压, 然后进入解压的目录下, 右键打开终端, 执行以下步骤:

(1) `./configure`



```
Configuration (LCM v1.3.1):  
  
Source code location: .  
Compiler:             gcc  
Java Support:          Enabled  
Python Support:        Disabled  
Lua Support:           Disabled
```

确保 java support is Enabled (操作 2 安装好后就会显示为 Enabled)

(2) `make`
(3) `sudo make install`
(4) `sudo ldconfig`

4 安装 eigen

(1) 解压, 在文件夹内执行
(2) `mkdir build`
(3) `cd build`
(4) `cmake ..`
(5) `make install`

5 安装 qt5.10

(1) 下载 `qt-opensource-linux-x64-5.10.0.run`


添加执行权限

(2) `chmod a+x qt-opensource-linux-x64-5.10.0.run`

(3) `./qt-opensource-linux-x64-5.10.0.run` 执行安装, (过程中需要邮箱账号, 也可以直接跳过注册页面执行下一步), 建议安装到用户的主目录下。

Qt 5.12.4 Setup

Qt Account - Your unified login to everything Qt



Please log in to Qt Account

Login Email

Password

[Forgot password?](#)

Need a Qt Account?

Sign-up Valid email address

Password

Confirm Password

☐ I accept the [service terms](#).

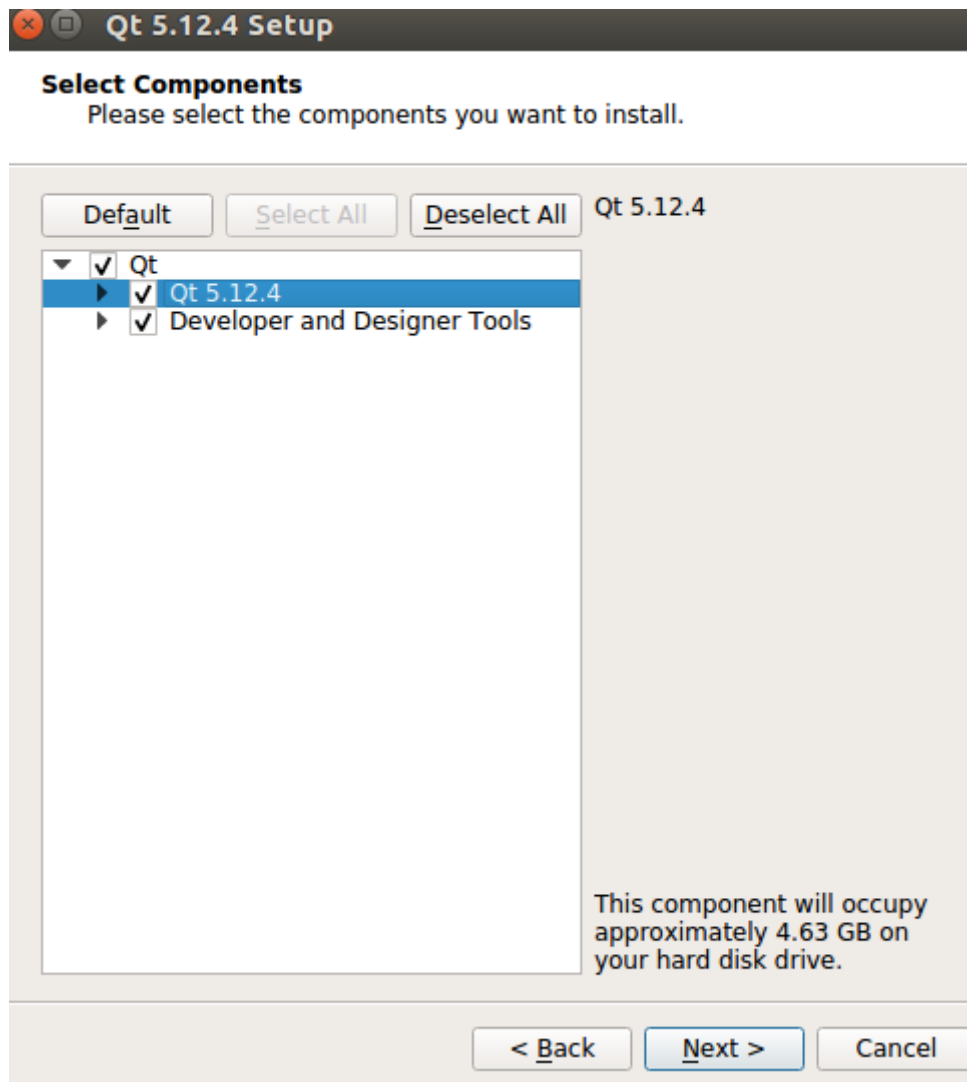
Settings < Back Skip Cancel

安装目录最后只保留 Qt，不要后面的 10.0，选择安装 Qt。

Installation Folder

Please specify the directory where Qt 5.12.4 will be installed.

Browse...



5. 安装 git

安装 git

指令: `sudo apt install git`

否则会出现下面错误

```
user@user-virtual-machine: ~/Downloads/Cheetah-Software-master/build
* ampl-interface , Interface to the AMPL solver library.
* slicot-interface , Interface to the controls library SLICOT.
* worhp-interface , Interface to the NLP solver Worhp (requires gfortran, gomp)
.
-- Downloading/updating googletest
CMake Error at /usr/share/cmake-3.5/Modules/ExternalProject.cmake:1757 (message)
:
  error: could not find git for clone of googletest-download
Call Stack (most recent call first):
  /usr/share/cmake-3.5/Modules/ExternalProject.cmake:2459 (_ep_add_download_command)
  CMakeLists.txt:9 (ExternalProject_Add)

-- Configuring incomplete, errors occurred!
See also "/home/user/Downloads/Cheetah-Software-master/build/googletest-download/CMakeFiles/CMakeOutput.log".
CMake Error at common/DownloadProject.cmake:171 (message):
  CMake step for googletest failed: 1
Call Stack (most recent call first):
  common/CMakeLists.txt:15 (download_project)
```

6. 可能出现问题

```
Reading package lists... Done  
E: Could not get lock /var/lib/apt/lists/lock - open (11: Resource temporarily  
unavailable)  
E: Unable to lock directory /var/lib/apt/lists/
```

解决: `sudo rm /var/lib/apt/lists/lock`

二、代码的编译与运行（全程需要联网）

1 编译

首先确保已经安装了 git

```
cd Cheetah-Software
```

```
cd scripts # 必须确保进入了该文件夹中
```

```
./make_types.sh #可能会看到报错 `rm: cannot remove...` 这是正常的
```

```
cd ..
```

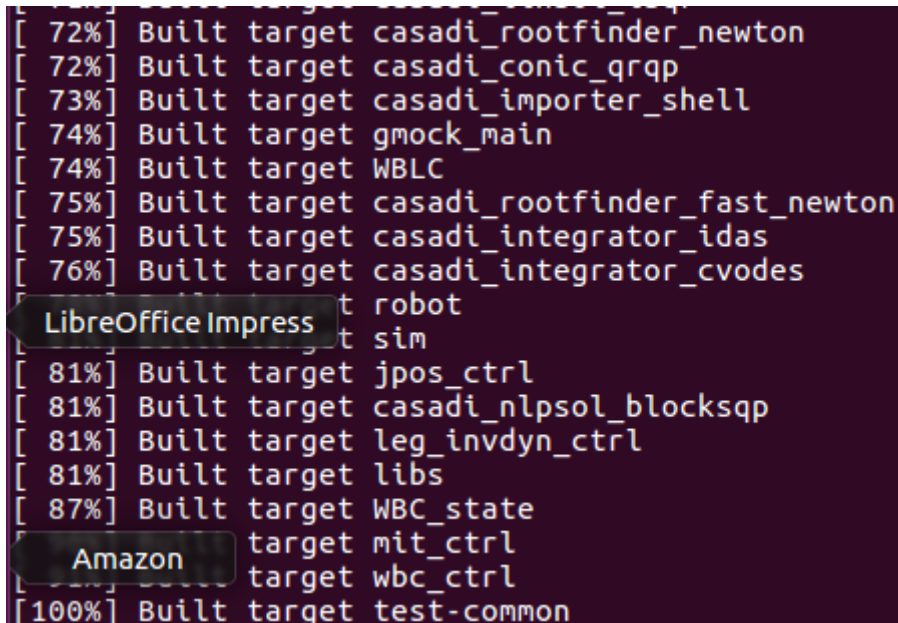
```
mkdir build
```

```
cd build
```

```
cmake -DMINI_CHEETAH_BUILD=TRUE ..
```

```
make -j4
```

（此过程与电脑配置有关，数字与电脑具备的线程数有关，根据实际情况选择）



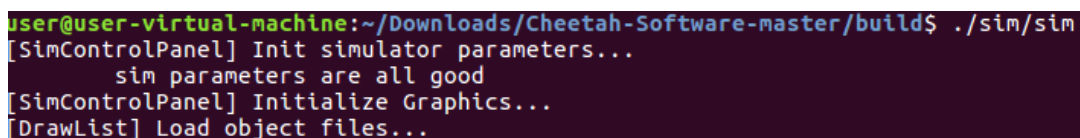
```
[ 72%] Built target casadi_rootfinder_newton
[ 72%] Built target casadi_conic_qrqp
[ 73%] Built target casadi_importer_shell
[ 74%] Built target gmock_main
[ 74%] Built target WBLC
[ 75%] Built target casadi_rootfinder_fast_newton
[ 75%] Built target casadi_integrator_idas
[ 76%] Built target casadi_integrator_cvodes
[ 76%] Built target robot
[ 76%] Built target sim
[ 81%] Built target jpos_ctrl
[ 81%] Built target casadi_nlpsol_blocksqp
[ 81%] Built target leg_invdyn_ctrl
[ 81%] Built target libs
[ 87%] Built target WBC_state
[ 87%] Built target mit_ctrl
[ 87%] Built target wbc_ctrl
[100%] Built target test-common
```

编辑完成

编译通过后既可以开始仿真、与机器人连接控制

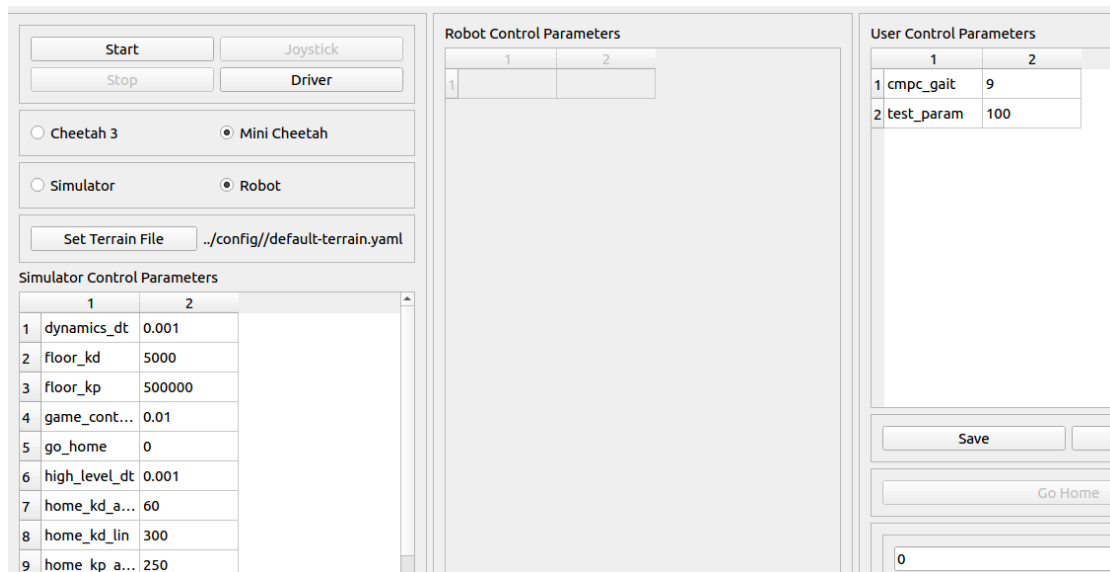
2 运行仿真

2.1 打开命令行窗口并进入/Cheetah-software/build 文件夹下，运行 `./sim/sim` 打开控制板



```
user@user-virtual-machine:~/Downloads/Cheetah-Software-master/build$ ./sim/sim
[SimControlPanel] Init simulator parameters...
sim parameters are all good
[SimControlPanel] Initialize Graphics...
[DrawList] Load object files...
```

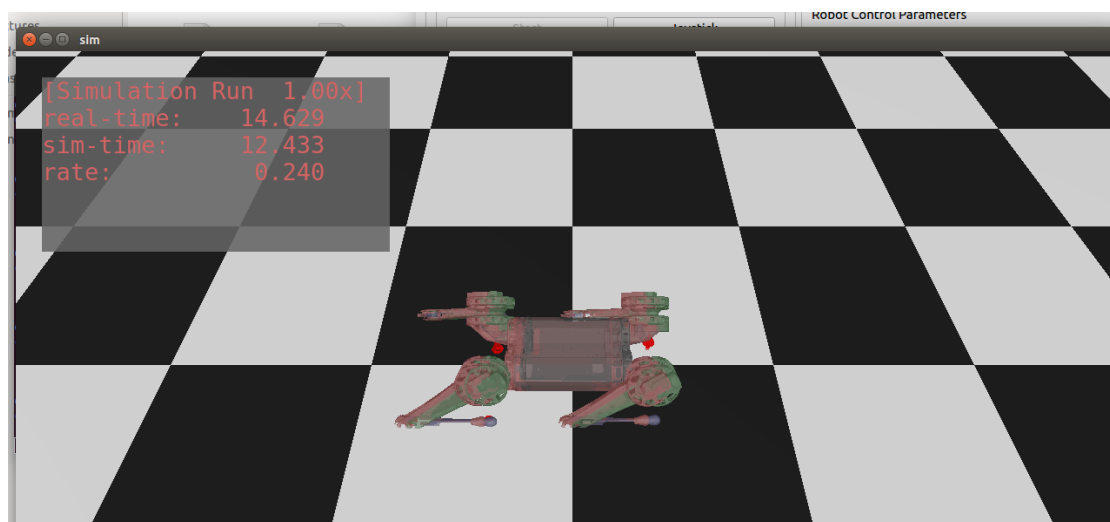
选中 Mini Cheetah 和 simulator（仿真必须先打开仿真界面 再运行程序 连接机器人时是先运行程序再打开控制界面选择）



2.2 在该目录下再打开新的终端窗口，运行 `./user/MIT_Controller/mit_ctrl m s`

```
user@user-virtual-machine:~/Downloads/Cheetah-Software-master/build$ ./user/MIT_Controller/mit_ctrl m s
[Quadruped] Cheetah Software
    Quadruped: Mini Cheetah
    Driver: Development Simulation Driver
[Shared Memory] open existing development-simulator size 266112 bytes
[Simulation Driver] Starting main loop...
Response(1) set robot s64 cheater_mode robot to: 0
Response(2) set robot double control_mode robot to: 0
Response(3) set robot double controller_dt robot to: 0.001
Response(4) set robot double foot_height_sensor_noise robot to: 0.001
Response(5) set robot double foot_process_noise_position robot to: 0.002
```

2.3 仿真开始运行



将 `use_rc` 设置为 0

将 `control_mode` 改为 6 (recovery stand)，运行到站立姿态

将 `control_mode` 改为 3 (balance stand)，平衡站立（力控状态）

将 `control_mode` 改为 4 (locomotion)，默认进行以 Tort 步态运行，此时可以用手柄操作

3 运行真实机器人

3.1 在主机电脑上的 Cheetah-software 目录下:

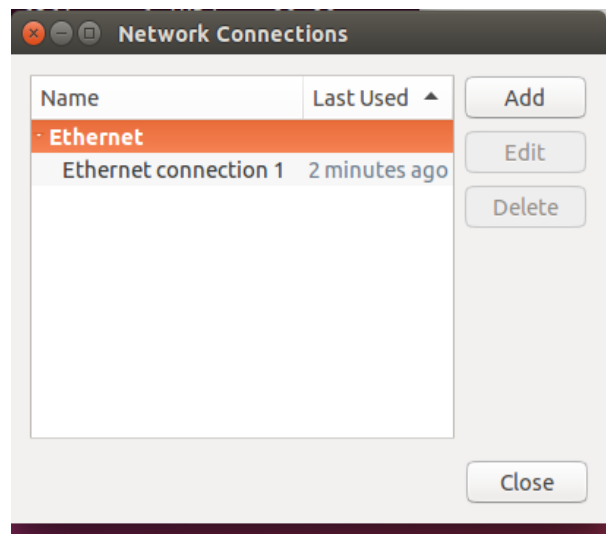
```
mkdir mc-build
```

3.2 在主机电脑上运行程序:

```
cd mc-build  
cmake -DMINI_CHEETAH_BUILD=TRUE ..  
make -j4
```

3.3 配置 ip

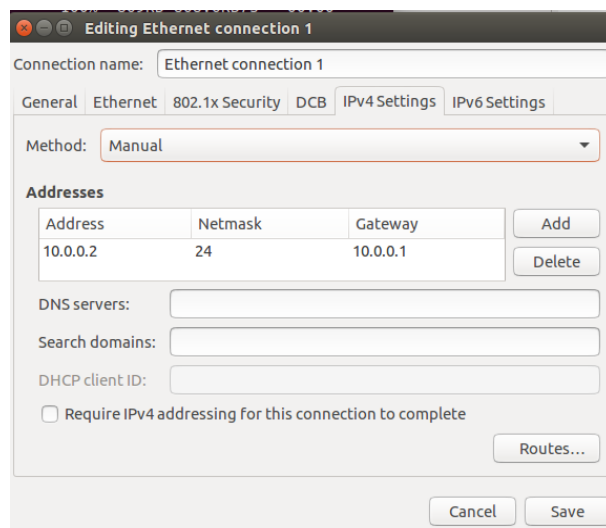
更改内部 Network Connections 添加一个新的 Ethernet
具体配置如下

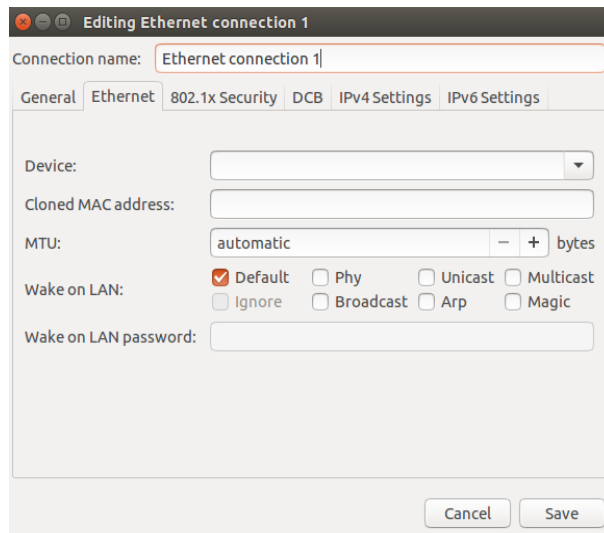


设置 Address:10.0.0.3

Netmask :255.255.255.0

Gateway: 10.0.0.1





设置完毕以后即可以与 upboard 进行通信

3.4 发送代码到 UP-Board 上

面: `../scripts/send_to_mini_cheetah.sh ./user/MIT_Controller/mit_ctrl`

```
bash: ../scripts/send_to_mini_cheetah.sh: No such file or directory
user@user-virtual-machine:~/Downloads/Cheetah-Software-master/mc-build$ ../scrip
ts/send_to_mini_cheetah.sh ./user/MIT_Controller/mit_ctrl
cp: './robot-software/build/librobot.so' and './robot-software/build/librobot.so
' are the same file
cp: './robot-software/build/libWBLC.so' and './robot-software/build/libWBLC.so'
are the same file
cp: './robot-software/build/libWBIC.so' and './robot-software/build/libWBIC.so'
are the same file
cp: './robot-software/build/libWBC_state.so' and './robot-software/build/libWBC_
state.so' are the same file
cp: './robot-software/build/libbiomimetics.so' and './robot-software/build/libbi
```

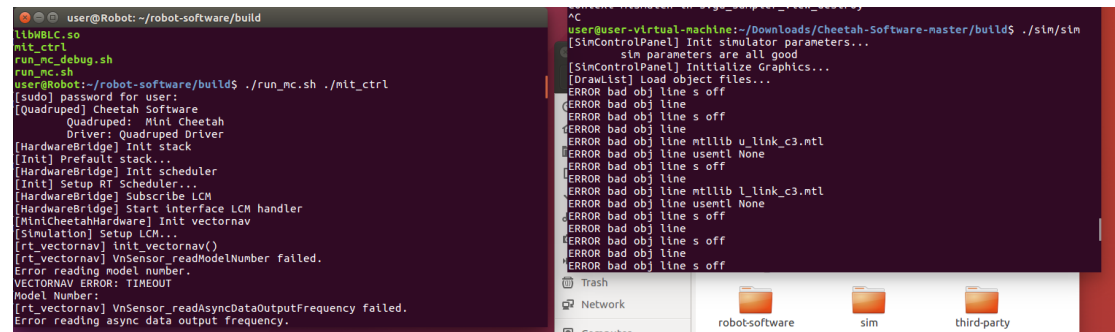
3.5 `ssh user@10.0.0.34`, 输入密码, 默认为: 123456

3.6 打开 UP-Board 文件夹: `cd ./robot-software/build`

3.7 运行 UP-Board 上面的程序: `./run_mc.sh ./mit_ctrl`

```
user@Robot: ~/robot-software/build
libWBLC.so
mit_ctrl
run_mc_debug.sh
run_mc.sh
user@Robot:~/robot-software/build$ ./run_mc.sh ./mit_ctrl
[sudo] password for user:
[Quadruped] Cheetah Software
    Quadruped: Mini Cheetah
    Driver: Quadruped Driver
[HardwareBridge] Init stack
[Init] Prefault stack...
[HardwareBridge] Init scheduler
[Init] Setup RT Scheduler...
[HardwareBridge] Subscribe LCM
[HardwareBridge] Start interface LCM handler
[MiniCheetahHardware] Init vectornav
[Simulation] Setup LCM...
[rt_vectornav] init_vectornav()
[rt_vectornav] VnSensor_readModelNumber failed.
Error reading model number.
VECTORNAV ERROR: TIMEOUT
Model Number:
[rt_vectornav] VnSensor_readAsyncDataOutputFrequency failed.
Error reading async data output frequency.
```


3.8 运行仿真器操作同仿真调试一致



真实连接界面如图，机器人位于地面下方一般为正确位置 在这个阶段可以观察图形界面可测试零位，俯仰角等是否正确

