

Hotpot Test (English version)

1. About yourself

1. Introduce yourself briefly and describe your past project experiences, including involvement in innovation projects, competitions, or course projects. Focus on the specific tasks you undertook, the knowledge you gained, and any lessons learned.
2. What is your purpose in joining the team? What do you hope to gain from participating in the RoboMaster (RM) Robotics Competition?
3. How much do you know about the RoboMaster (RM) Robotics Competition? You can watch recent competition videos or documentaries on Bilibili. Share your understanding of the competition and discuss your strengths.
4. Team members often have different opinions and conflicts. How would you resolve conflicts with other team members when they arise?

2. Technical Questions

1. Install Ubuntu

Install the Ubuntu 22.04 LTS distribution system on your own; it is recommended to set up a dual-boot system. Note: Do NOT install a Ubuntu virtual system, you need to run your code in a dual-boot system in this test.

Be familiar with basic commands in the Linux terminal, such as `cd`, `vim`, `apt`, `rm`, `cp`, `mv`, `wget`, `curl`, `cat`, `ls`, `pwd`, `service`, etc. **Take screenshots to document the effects of each command.**

2. Network

Maybe for some reasons you can't access Google or Chatgpt directly on the Internet. You should configure the network in your computer accurately and try to access Google or Chatgpt. You will need them later.

3. ROS2 System

Install ROS2 Humble robotic operating system in your Ubuntu, read the official ROS2 documentation to learn basic ROS2 knowledge. Refer to official examples and perform basic ROS2 (C++) operations, such as creating a workspace, writing and publishing nodes, listening to nodes, etc. **Capture screenshots of the results and submit them.**

[ROS2 Humble Document](#)

4. ROS2 Navigation2

Learn with the simulated robot TurtleBot3 or TurtleBot4, refer to the official documentation, and hands-on reproduce some examples. **Record the process and submit the video.**

[Nav2 Document](#)

[Other Document](#)

5. OpenCV

Refer to the official OpenCV documentation, configure and install OpenCV by yourself, and use OpenCV-C++ to implement some basic image processing operations such as binarization, dilation, erosion, filtering, edge detection, etc. **Capture screenshots of the results and submit the source code.**

[OpenCV: OpenCV Tutorials](#)

6. Kalman Filter

Watch the linked documents and videos (or you can search on Youtube), learn about Kalman filtering, and **take detailed study notes to document your understanding.**

[Kalman Filter Tutorial](#)

[从放弃到精通！卡尔曼滤波从理论到实践~哔哩哔哩bilibili](#)

7. Camera Model

Watch the linked videos, study camera model-related knowledge, and **take thorough study notes to document your understanding in the document.**

[11.三维重建-摄像机模型.1080P 哔哩哔哩bilibili](#)

8. CMake

Refer to the CMake documentation and complete all eleven examples in the first section of the basic tutorial. **Capture screenshots and submit them.**

[前言 · GitBook \(sfumecjf.github.io\)](#). You can also search for some other English documents.

9. Git

Install Git, create your own GitHub repository, and perform some basic Git operations. Focus on tasks such as establishing a Git repository, cloning a repository, committing and pushing files, pulling changes, and branching. **Capture screenshots and submit them.**

10. Pytorch

Configure the PyTorch deep learning library on Ubuntu according to the PyTorch Chinese documentation(or other docs), and recommend using miniconda for Python environment management.

Based on the MNIST handwritten digit dataset, choose any model approach to implement handwritten digit recognition with an accuracy of over 90%. **Please submit the source code (with explanatory comments) and screenshots of the results.**

Open Source Vision Learning Materials (Thanks to senior members of the vision team.)

<http://alist.geekbang.top/>

Welcom to ask any questions.