The hardware platform used in this experiment is the FS-AIROBOTB intelligent robot. The hardware composition is shown in Fig 8, and the description of each component is shown in Table 7.

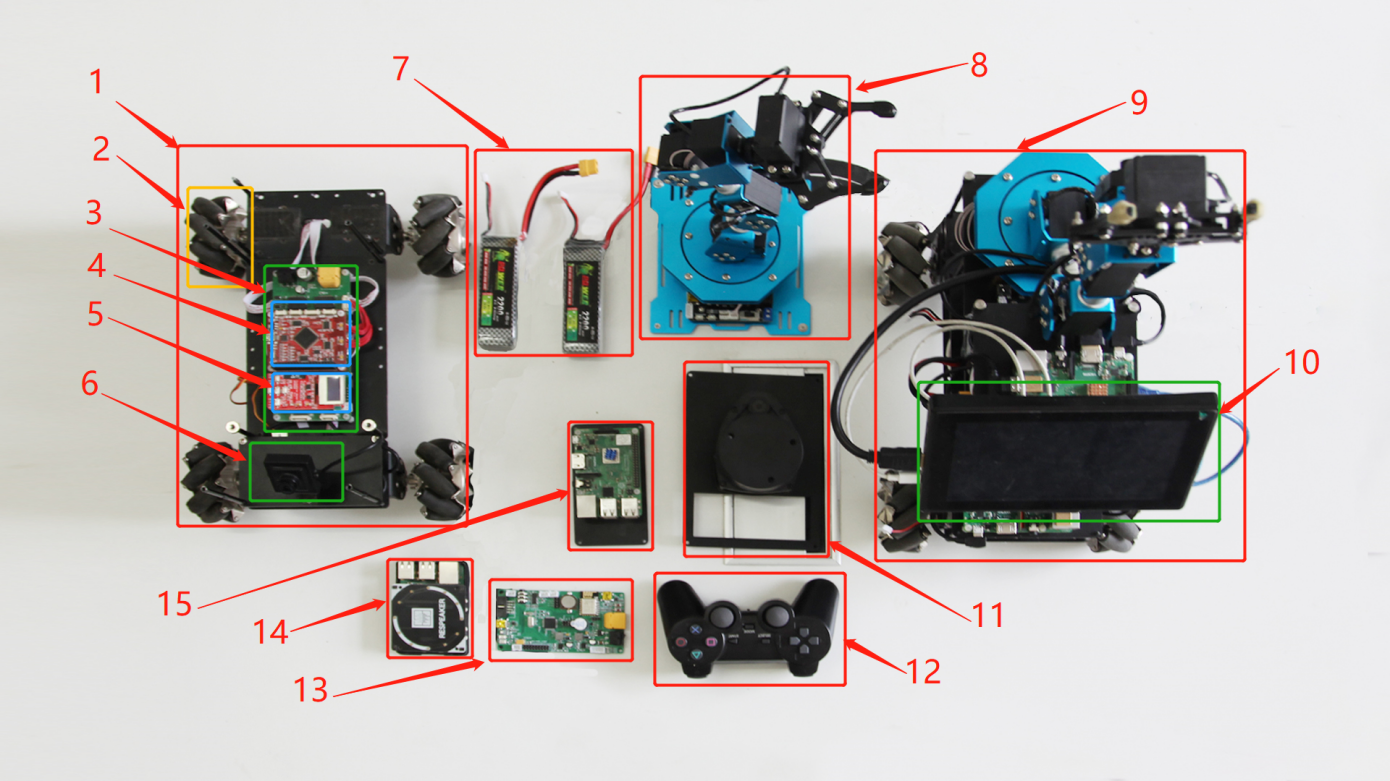


Fig 8. FS-AIROBOTB component diagram

Table 7 FS-AIROBOTB component serial number comparison table

|  |  |  |  |
| --- | --- | --- | --- |
| Serial number | Part name | Serial number | Part name |
| 1 | ROS omnidirectional vehicle chassis | 9 | FS\_AIROBOTB |
| 2 | Mecanum wheel | 10 | 7 inch HDMI display |
| 3 | Omnidirectional vehicle drive module | 11 | 360 degree lidar |
| 4 | Cortex-M4 chassis core control board | 12 | Wireless bluetooth remote control handle |
| 5 | FS\_Explore sensor board | 13 | Cortex-M3 robotic arm control board |
| 6 | 1080P industrial module camera | 14 | 4 array microphones |
| 7 | Grep 3S/25C/1300mA power lithium battery | 15 | 3B+ Raspberry Pi |
| 8 | Six degrees of freedom robotic arm |  |  |

To test the effectiveness of the improved A\* algorithm, it is transplanted to the FS-AIROBOTB mobile robot hardware platform in this paper. Given the open-source nature of the ROS and the fact that the ROS contains the navigation function package "nav\_ROS" for path planning, the original algorithm of the function package is rewritten to realize the transplant of the algorithm. This method can achieve the advantages of rapid development and tighter integration with other components of the algorithm.

This experiment is divided into three steps; code transplantation, map construction and autonomous navigation testing.

1. Code transplantation. The main steps are as follows;

1) Use Virtual Network Console (VNC) to connect the FS\_AIROBOTB robot and enable the PC to communicate with the robot.

2) Modify the source code and write the EBS-A\* algorithm into the code file, specifically Astar.cpp, planner\_core.cpp and grid\_path.cpp. The programming language is C++.

3) Compile the modified source code.

4) After the compilation is complete, execute the ROSLaunch command to restart the ROS.

This completes the porting of the code to the FS\_AIROBOTB.

1. Map construction. The main steps are as follows;
2. Modify the IP address of the ROS robot.
3. Use VNC to connect to the ROS robot.
4. Enter the function folder of the ROS robot and execute the my\_bringup\_1.sh script to start the robot construction, radar SLAM and camera functions.
5. After step 3 is completed, start the robot visualizer (rviz) software to realize map scanning, complete map construction and save the map.
6. Robot autonomous navigation test. The main steps are as follows;

1) Start the robot construction, navigation script and camera function services in the my\_bringup\_1.sh script.

2) Step 1 After completion, start the rviz software and open the navigation work file navigation.rviz in it.

3) Select any target point on the map, and the robot will plan a smooth curve without collision at this time, as shown in Fig 9.