# Ultrasonic follow



This tutorial is a comprehensive experiment combining multiple peripherals. You can first understand a single peripheral before performing this experiment.

## 1. Software-Hardware

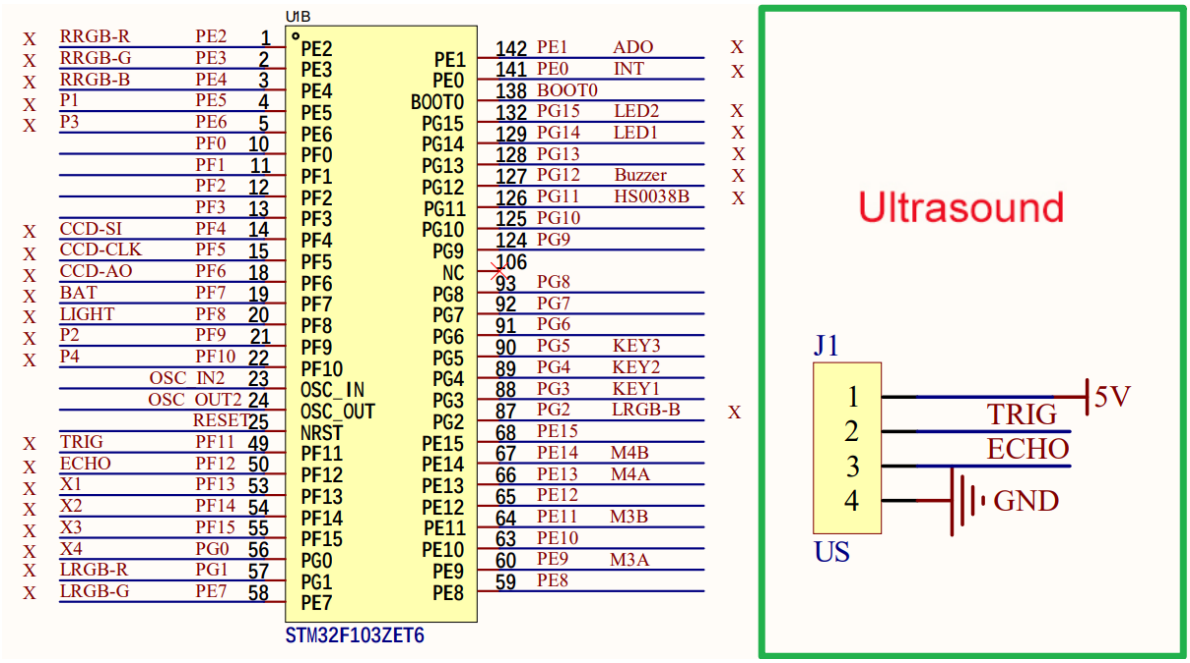
- **STM32F103CubeIDE**
- **STM32 Robot Development Board**

  Ultrasonic module, 310 motor*4: external
- **Type-C data cable or ST-Link**

  Download programs or simulate the development board

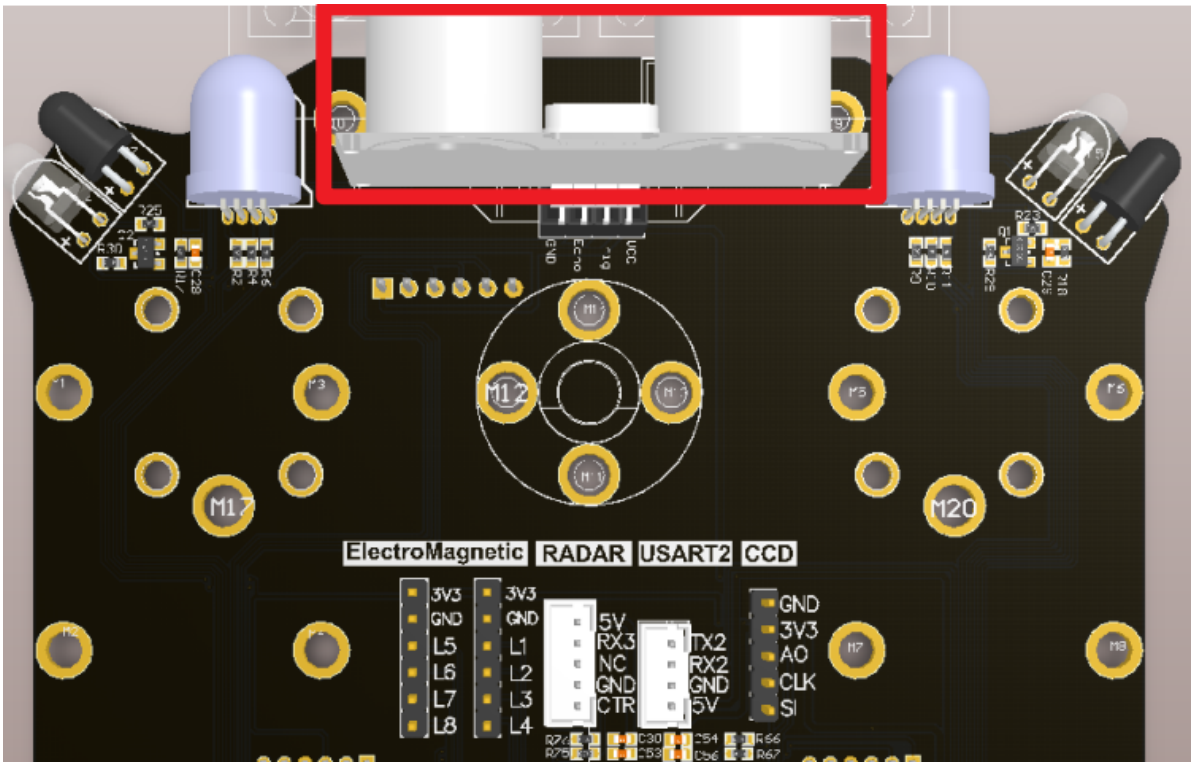
## 2. Brief principle

### 1. Hardware schematic diagram

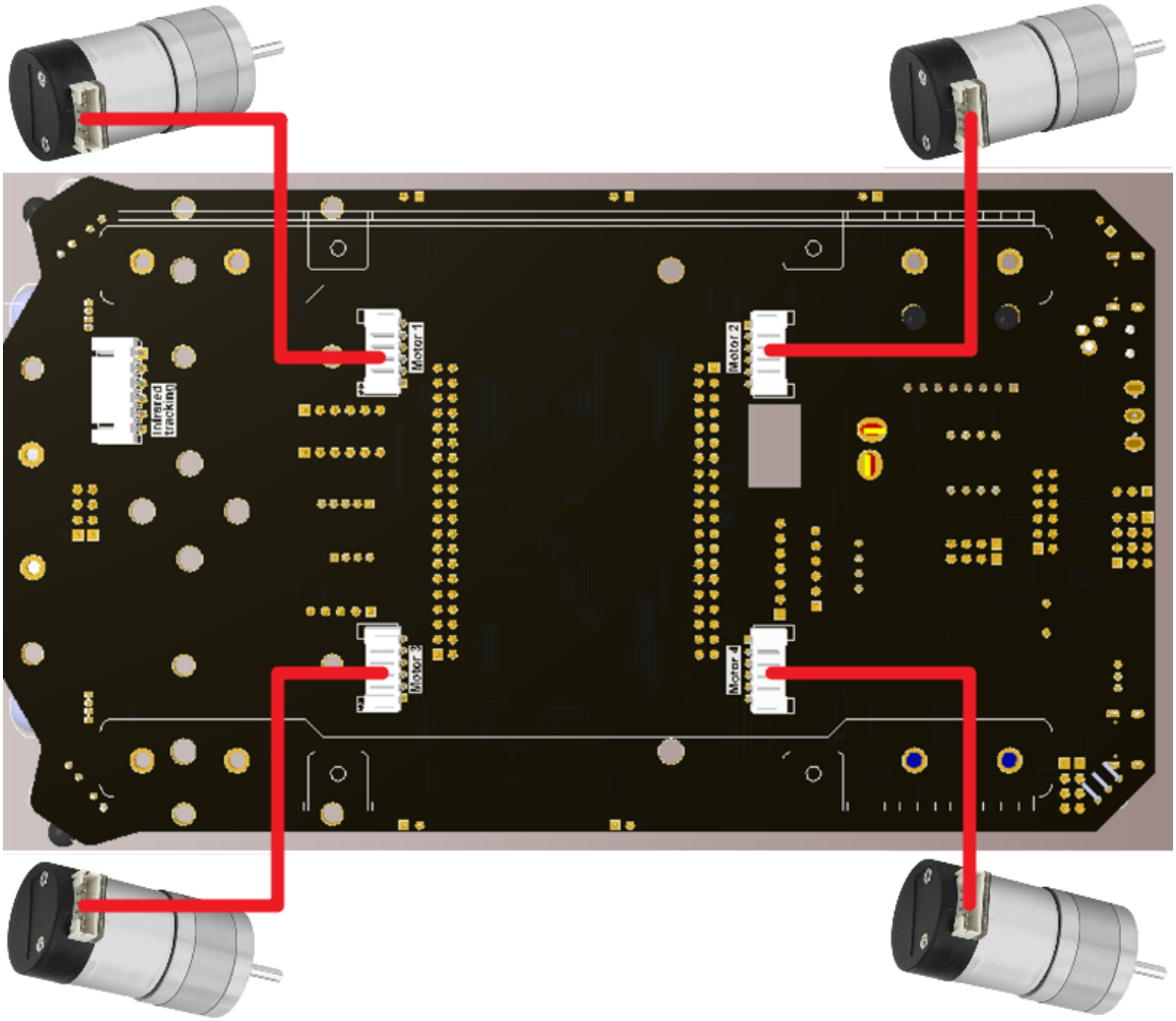


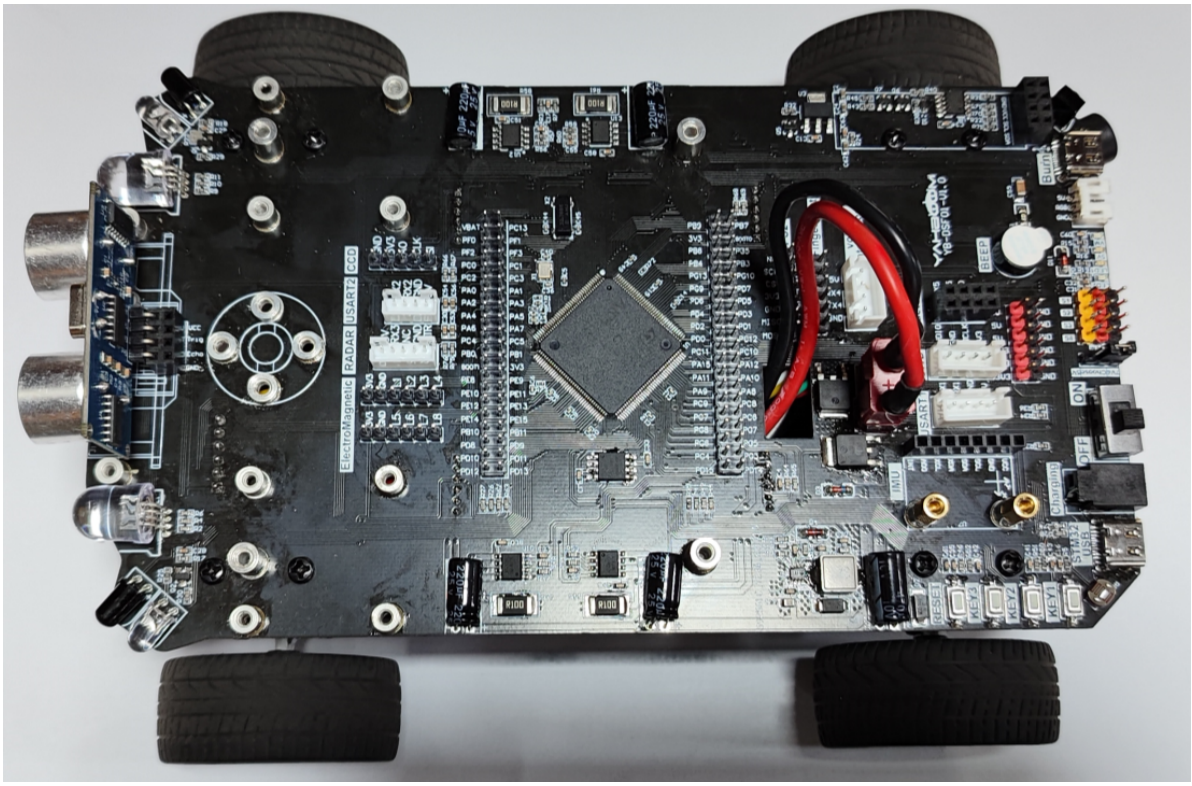
## 2. Physical connection diagram

- **Ultrasonic Installation**




- **Motor Wiring**





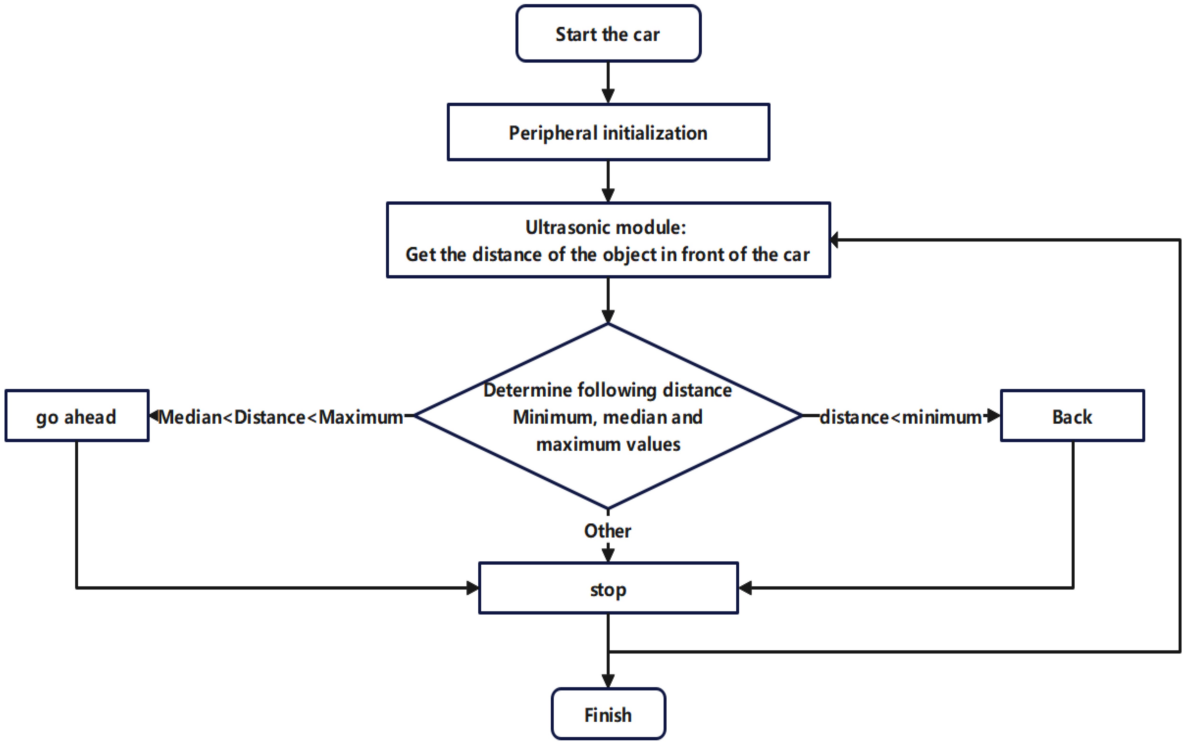

```
For battery pack assembly, please see [5. Smart Car Experiment: Car Line Patrol]
```

## 3. Control principle

The distance of the object in front is obtained through the ultrasonic module, and the car is controlled to move forward and backward according to the distance.

```
The ultrasonic module can obtain the distance of the object in front, and we
judge the position of the car based on this distance:
If the car is close to the object, we call the car's backward function in the
corresponding judgment function;
If the car is far away from the object, we call the function of the car moving
forward in the corresponding judgment function;
If the car happens to be at the distance we set, we call the car stop function in
the corresponding judgment function.
```

- **Program flow chart**




| External modules | Functions |
| --- | --- |
| Ultrasonic module | External information collection: Obtain object distance |
| Motor | Motion Control |

# 3. Main functions

**This tutorial does not use PID to control the movement of the car**

```
The functions introduced before will not be introduced again!
```

> **Function: Ultrasonic_follow**

| Function prototype | void Ultrasonic_follow(uint16_t Max_distance,uint16_t Min_distance) |
| --- | --- |
| Function description | Ultrasonic following |
| Input parameter 1 | **Max_distance**: follow the maximum distance |
| Input parameter 2 | **Min_distance**: follow the minimum distance |
| Return value | None |

> **Function: Get_distance**

| Function prototype | float Get_distance(void) |
| --- | --- |
| Function description | Ultrasonic ranging |
| Input parameters | None |

| Function prototype | float Get_distance(void) |
| --- | --- |
| Return value | Distance (five times average) |

> **Function: wheel_State**

| Function prototype | void wheel_State(uint8_t state, uint16_t speed) |
| --- | --- |
| Function description | Control car movement |
| Input parameter 1 | **state**: Car status |
| Input parameter 2 | **speed**: speed control |
| Return value | None |

```
For the underlying driver, you can refer to the tutorials in Chapter 3 and
Chapter 4.
For the application layer, you can read the source code in the project file
yourself
```

# 4. Experimental Phenomenon

After successfully downloading the program, press the RESET button on the development board to observe the effect of the car!

```
For program download, please refer to [2. Development environment construction
and use: program download and simulation]
```

**Phenomenon:**

**Hand close to the ultrasonic module**: The car moves backwards

**Hands away from the ultrasonic module**: The car moves forward. When the distance exceeds the maximum following value, the car stops.

**Keep your hand within the following distance**: The car will move forward first and then stop.

```
You can see the experimental phenomenon [Ultrasound Follow_Experimental
Phenomenon.mp4]
```