Radar obstacle avoidance

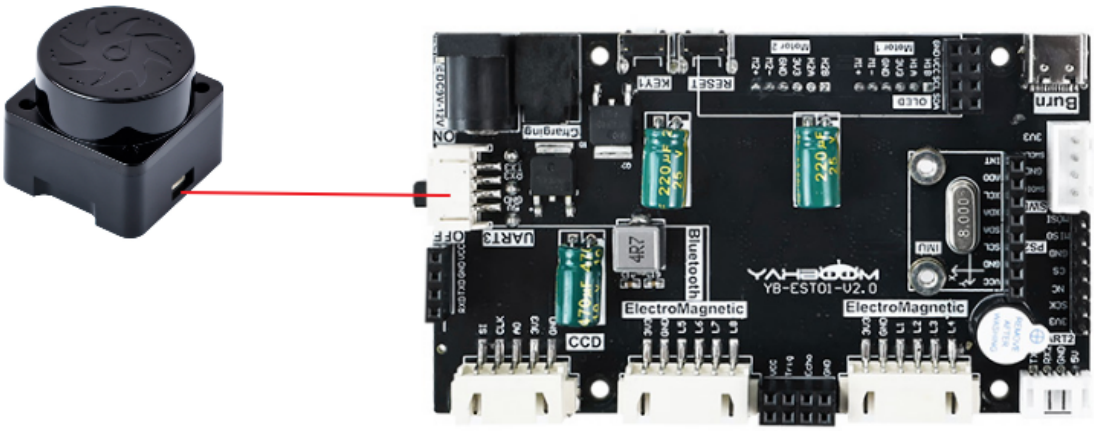
The tutorial mainly demonstrates the obstacle avoidance function of the balance car combined with the Tmini-Plus radar.

```
The tutorial only introduces the standard library project code
```

Hardware connection

Since we have configured a special connection line, you only need to install it to the corresponding interface.

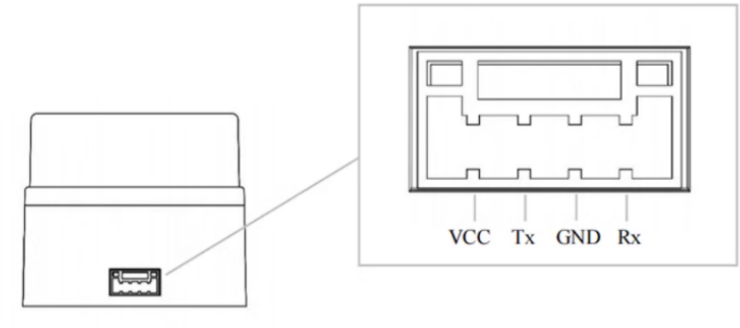
Peripherals	Development Board
Tmini-Plus radar: VCC	5V
Tmini-Plus radar: TXD	PC10
Tmini-Plus radar: RXD	PC11
Tmini-Plus radar: GND	GND



Control principle

The program analyzes the radar data and determines the direction of movement of the balance car based on the distance information at the specified angle.

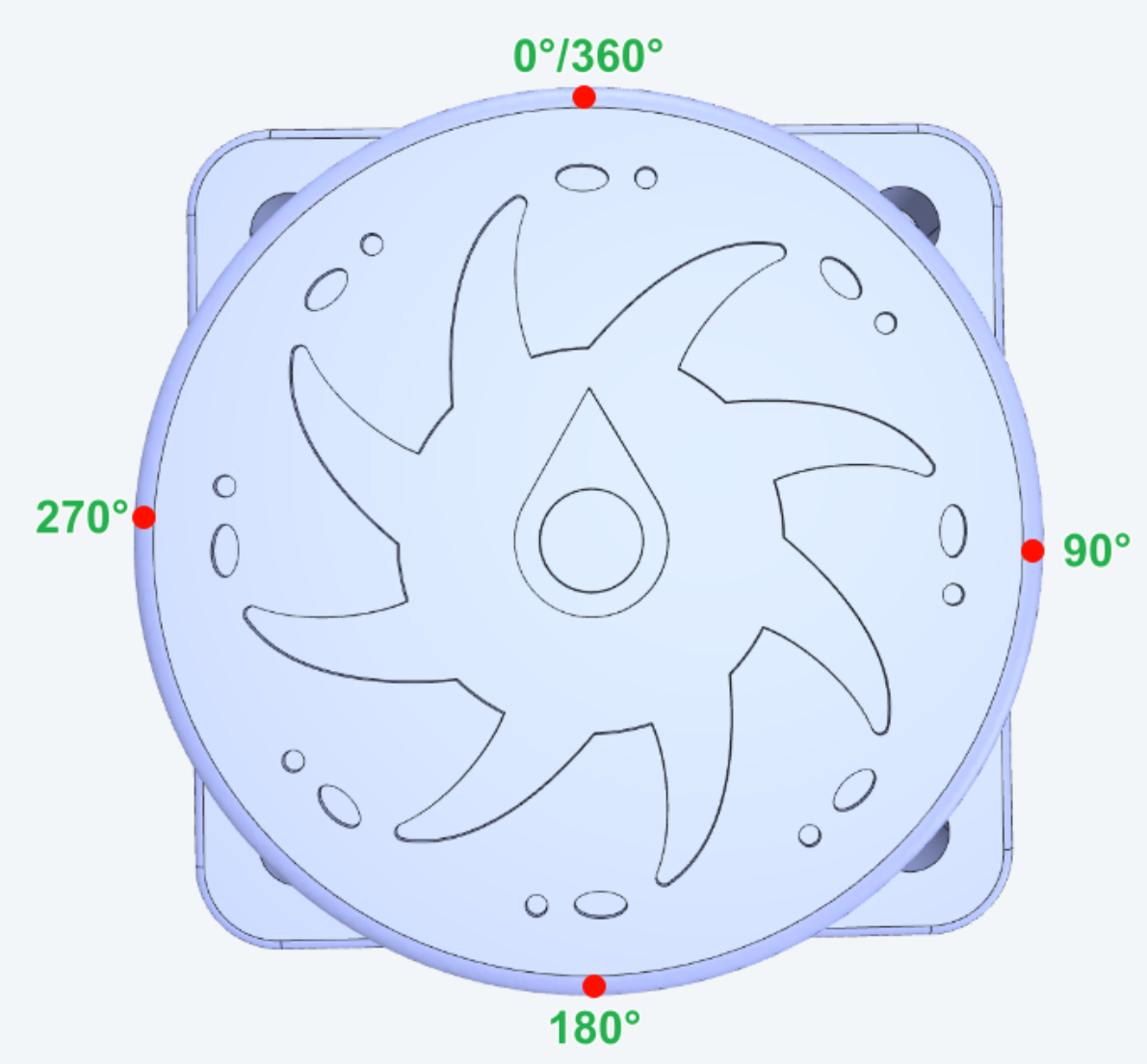
- Tmini-Plus radar



Product name	Tmini-Plus radar
Scanning frequency	6-12Hz
Sampling frequency	4000 times/s
Measuring radius	Black object: 12m
Minimum measuring distance	0.05m
Ranging principle	TOF ranging
Scanning angle	360°
Communication interface	Standard asynchronous serial port (UART) 1. Baud rate: 230400 2. Data bits: 8 3. Check bit: None 4. Stop bit: 1
ROS support	ROS1/ROS2
Windows support	Host computer

> **Radar Angle Distribution**

The arrow in the center of the radar points to 0°/360°, and the angle increases clockwise.



> **Communication Protocol**

```
For detailed information, please refer to the "T_Mini_Plus Manual"
```

Main Code

The tutorial mainly explains the code for the radar obstacle avoidance function. For detailed code, please refer to the corresponding project file.

> **Car_Avoid**

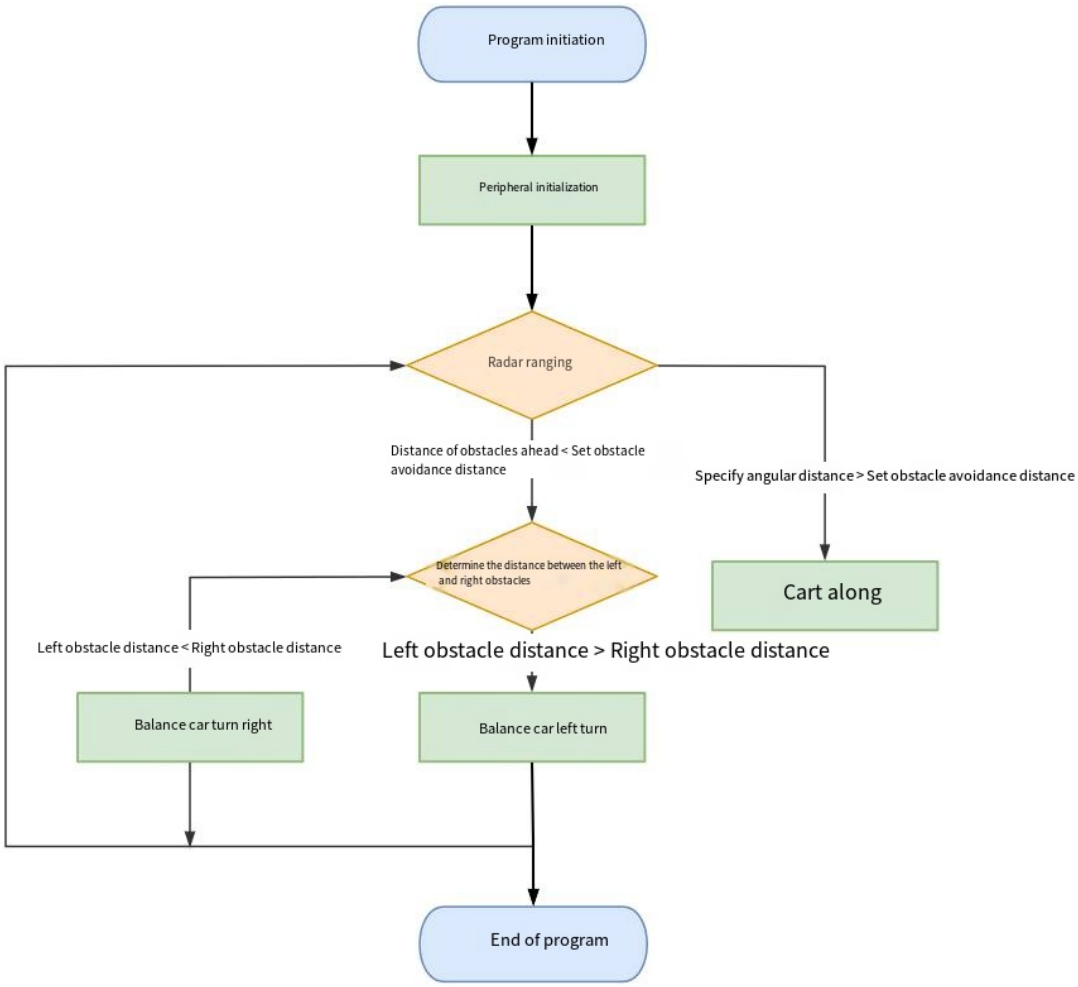
The distance data of the radar at 0°, 90°, and 270° is used to determine the obstacle position of the balance car and control the movement of the balance car.

```
void Car_Avoid(void)
{
    static u8 avoid_step = 0;//Obstacle Avoidance Steps

    //Each time this function is called, the value is dynamically refreshed.
    float get_data_mid = Tminidis[0];
    float get_data_Lmid = Tminidis[270];//Left
    float get_data_Rmid = Tminidis[90];////Right

    switch(avoid_step)
    {
```

```
        case 0:
        if(get_data_mid <avoid_dis && get_data_mid >0 )//Obstacle Avoidance
        {
            //Car stop
            Move_X = 0,Move_Z = 0;
            delay_time_int(10);
            avoid_step = 1;
        }
        else
        {
            //Car moving forward
            Move_X = GO_speed,Move_Z = 0;
        }
        break;
        case 1: if(get_time_int()==0) avoid_step ++;break; //Time to go to next
step
        case 2:
        {
            //Trolley backward
            Move_X = -15,Move_Z = 0;
            delay_time_int(100); //1s
            avoid_step = 3;
        }
        case 3: if(get_time_int()==0) avoid_step ++;break; //Time to go to next
step
        case 4:
        {
            if(get_data_Lmid >= get_data_Rmid )
            {
                //left
                Move_X = 0,Move_Z = -450;
                delay_time_int(100);
            }
            else
            {
                //right
                Move_X = 0,Move_Z = 450;
                delay_time_int(100);
            }
            avoid_step = 5;
        }
        case 5: if(get_time_int()==0) avoid_step=0;break; //Time to go to next
step
    }
}
```

Program flow chart

Briefly introduce the process of function implementation:



Experimental phenomenon

Software code

The Balance_Radar_Avoid.hex file generated by the project compilation is located in the OBJ folder of the Balance_Radar_Avoid project. Find the Balance_Radar_Avoid.hex file corresponding to the project and use the FlyMcu software to download the program into the development board.

```
Product supporting data source code path: Attachment → Source code summary → 5.Balanced_Car_Extended → 12.Balance_Radar_Avoid
```

Experimental phenomenon

After the program is started, press KEY1 according to the OLED prompt to start the radar obstacle avoidance function of the balance car: OLED displays start control!

Radar does not detect obstacles (obstacle distance>200mm): the car moves forward

Radar detects obstacles in front (0mm<obstacle distance<200mm): the car stops and moves backward

Judge the distance of left and right obstacles:

Left obstacle distance ≥ right obstacle distance: the car turns left and then moves forward

Left obstacle distance <right obstacle distance: the car turns right and then moves forward

The program has voltage detection. If the voltage is less than 9.6V, a low voltage alarm is triggered and the buzzer will sound.
Common situations that trigger voltage alarms:
1. The power switch of the development board is not turned on, and only the Type-C data cable is connected for power supply
2. The battery pack voltage is lower than 9.6V and needs to be charged in time