Lidar guard

The tutorial mainly demonstrates the guard 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, we only need to install it to the corresponding interface.

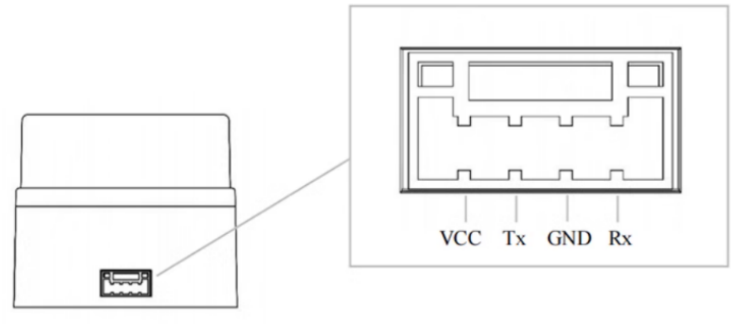
Peripherals	Development Board
Tmini-Plus radar: VCC	5V
Tmini-Plus radar: TXD	PC10
Tmini-Plus radar: RXD	PC11
Tmini-Plus radar: GND	GND

Control principle

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

```
The minimum distance of the specified angle collected controls whether to issue an alarm; the angle corresponding to the minimum distance controls the rotation direction of the radar
```

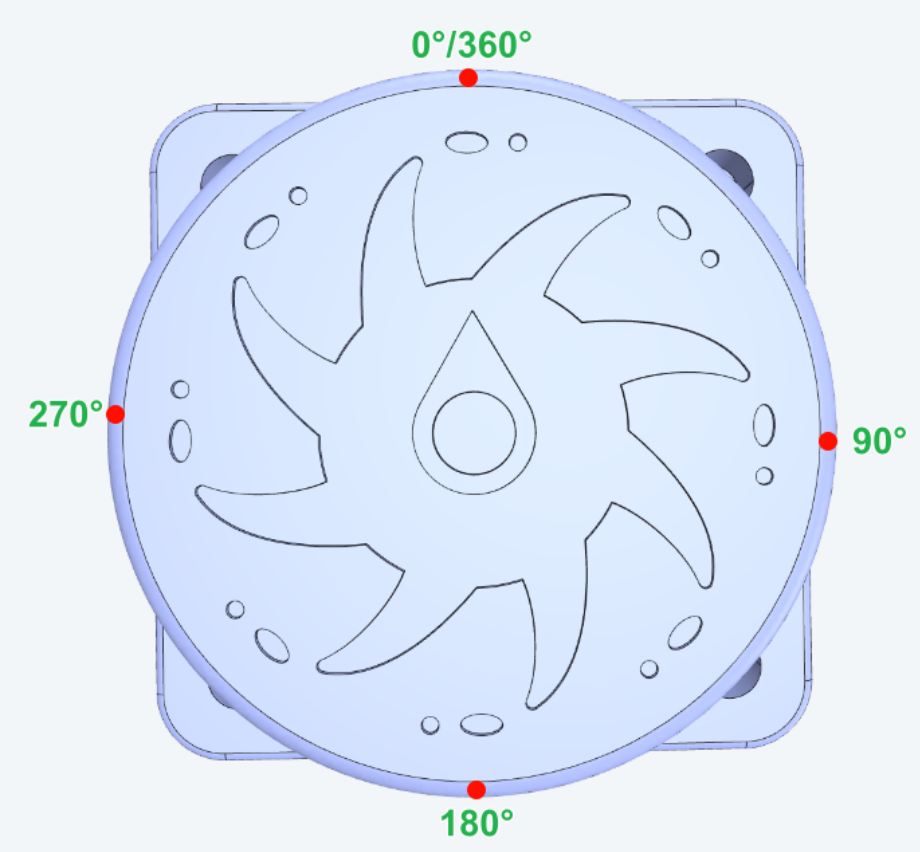
- Tmini-Plus radar



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 position: 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 implementing the radar guard function. For detailed code, please refer to the corresponding project file.

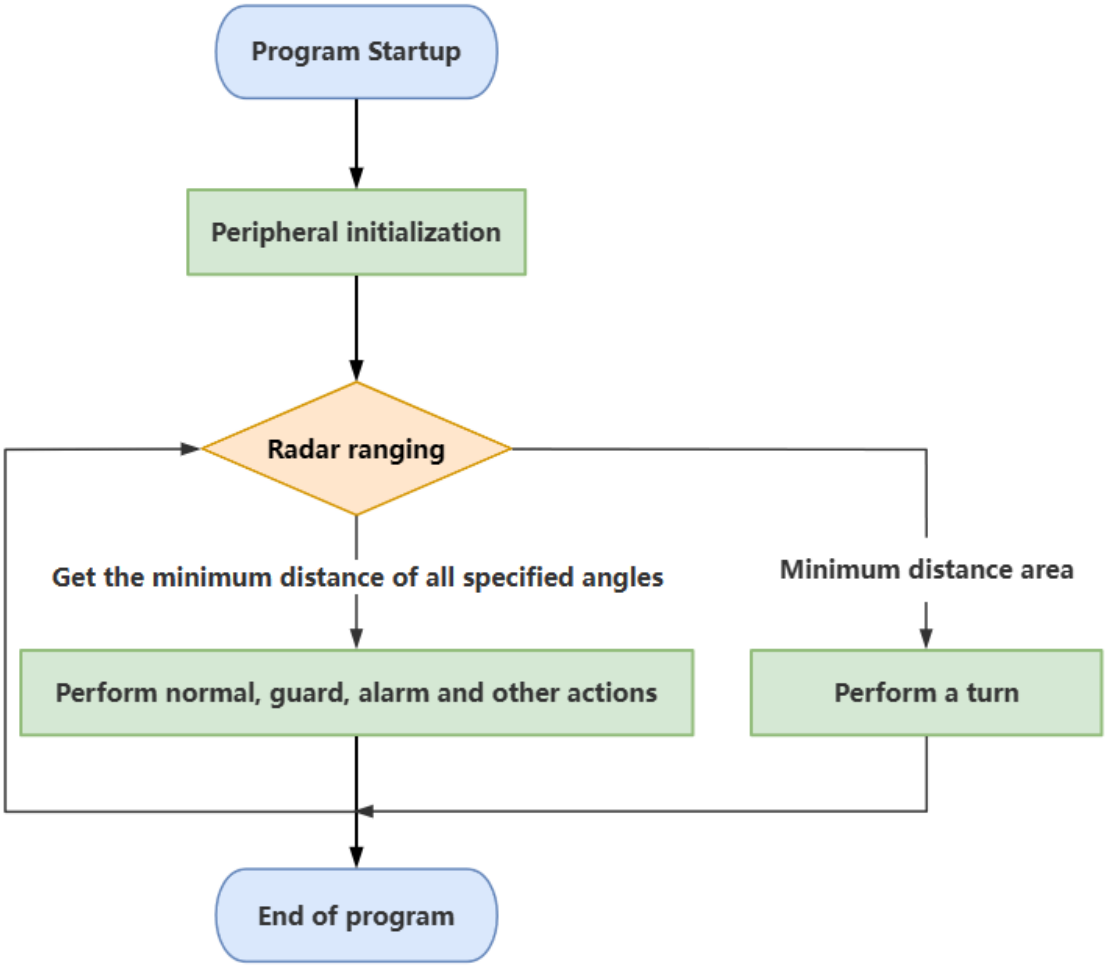
> **Car_Alarm**

The buzzer switch is controlled by reading the minimum distance data of the radar specified angle, and the angle corresponding to the minimum distance controls the steering of the balance car.

```
void Car_Alarm(void)
{
    Get_DIS_Group();

    if(data_min >= decect_Max_dis) // Exceeding the maximum distance
    {
        Move_Z = 0; // Car stop
        BEEP_OFF;
        return;
    }

    if( data_min <= BEEP_Alarm_Dis)
    {
        BEEP_ON;
    }
    else if( data_min <= Alarm_Dis)
    {
        BEEP_OFF;
    }
    else // Not in range
    {
        BEEP_OFF;
        Move_Z = 0;// Car stop
    }

    // Control car
    if(data_min_index<3) // 3 is the middle value
    {
        Move_Z = -450; // The car turns left
    }
    else if(data_min_index>3)
    {
        Move_Z = 450;// The car turns right
    }
    else
    {
        Move_Z = 0;
    }
}
```

Program flow chart

Briefly introduce the process of function implementation:



Experimental phenomenon

> **Software code**

The Balance_Radar_Alarm.hex file generated by the project compilation is located in the OBJ folder of the Balance_Radar_Alarm project. Find the Balance_Radar_Alarm.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 → 13.Balance_Radar_Alarm
```

> **Experimental phenomenon**

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

The radar detects an object in the front area (the minimum angle distance obtained ≥ the maximum guard distance (400mm)): the car stops and maintains a balanced state;

The radar detects an object in the front area (the minimum angle distance obtained ≤ the alarm distance (300mm)): the buzzer sounds;

The radar detects an object in the front area (the minimum angle distance obtained ≤ the guard distance (400mm)): the car stops and maintains a balanced state;

The radar detects an object in the front area (the minimum angle distance obtained corresponds to the angle position): located in the upper left, the car turns left; located in the upper right, the car turns right.

The program has voltage detection. If the voltage is less than 9.6V, a low voltage alarm is triggered and the buzzer sounds.
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