

Lidar patrol

Lidar patrol

- Hardware connection
- Control principle
- Main code
- Program flow chart
- Experimental phenomenon

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

The tutorial only introduces the standard library project code

Hardware connection

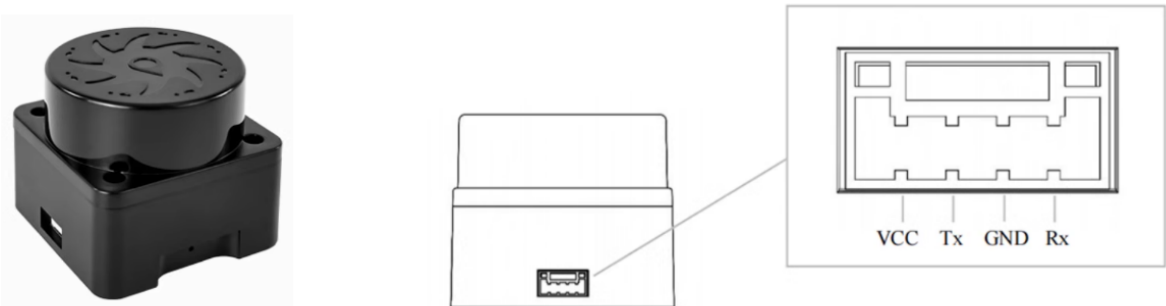
Since we have configured a special connection line, we only need to install it to the corresponding interface.

| 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 whether there is an obstacle directly in front of the patrol route based on the 0° distance.

- Tmini-Plus radar

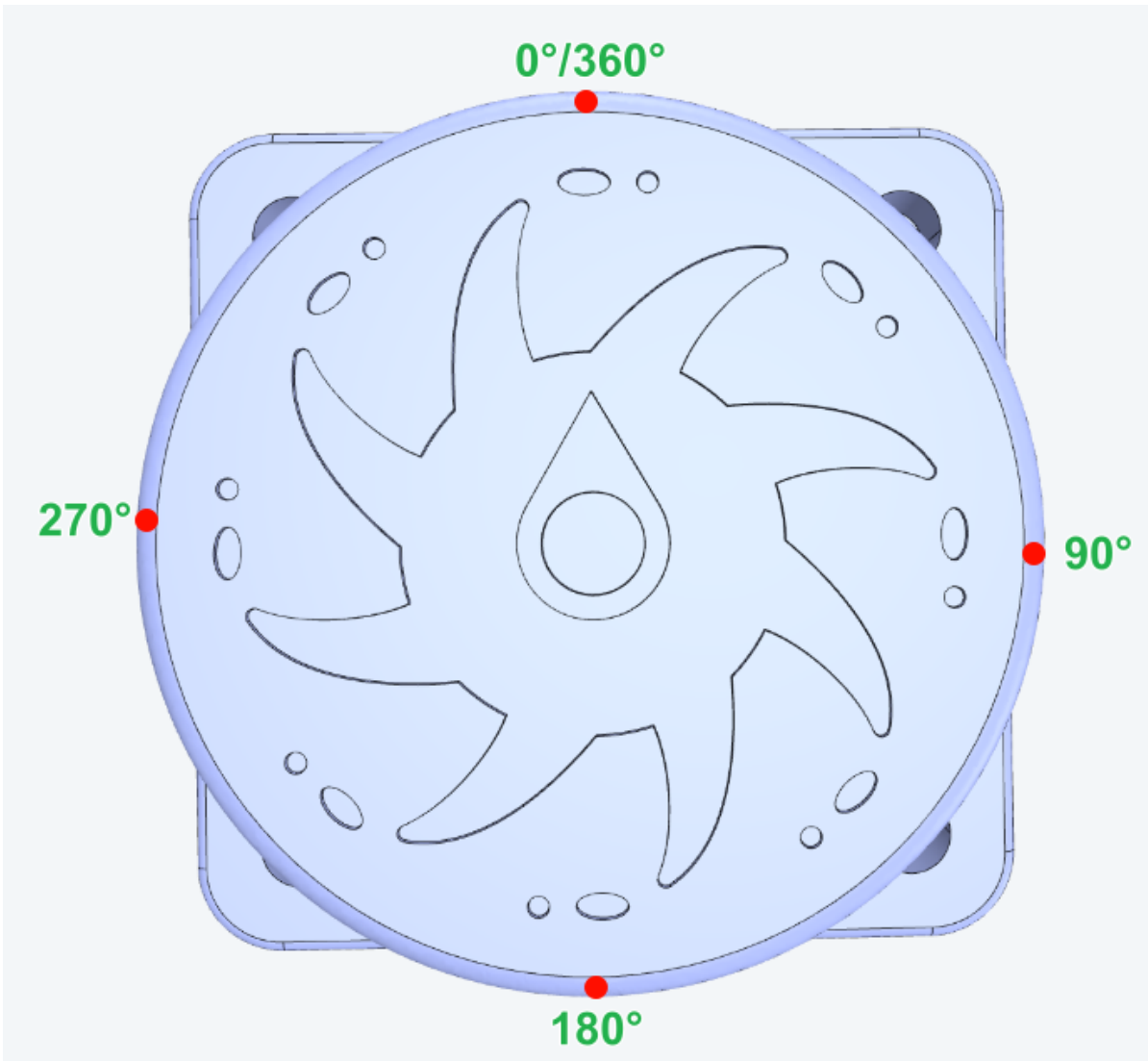


| Product name | Tmini-Plus radar |
|--------------------|------------------|
| Scanning frequency | 6-12Hz |
| Sampling frequency | 4000 times/s |

| | |
|----------------------------|---|
| Product name | Tmini-Plus radar |
| 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

Main code

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

Car_Patrol

The radar 0° distance data is used to determine whether there is an obstacle in front of the balance car, and the patrol function is realized through the set action.

```
void Car_Patrol(void)
{
    static u8 patrol_step = 0;
    static u8 runtimes = 1;

    float Patroldis = Tminidis[0]; // Only handles 0° angles

    if(patrol_step >= 7)
    {
        // Patrol function has stopped
        return;
    }

    // when there are obstacles on the patrol path
    if(Patroldis < 200 && Patroldis > 0)
    {
        // The following operations are performed only once
        if(stop_patrol == 1)
        {
            BEEP_ON;
            Move_X_old = Move_X; // Backup
            Move_Z_old = Move_Z; // Backup
            Move_X = 0, Move_Z = 0; // parking
            stop_patrol = 0; // Pause timer
        }
        return;
    }

    if(stop_patrol == 0)
    {
        Move_X = Move_X_old; // recover
        Move_Z = Move_Z_old; // recover
        BEEP_OFF;
        stop_patrol = 1; // Timing
        return;
    }

    switch(patrol_step)
    {
        case 0: myTurn_Kd = string_pd, Move_X = 15; Move_Z = 0;
        delay_time_int(Go_time); patrol_step++; break; // go ahead
        case 1: if(get_time_int() == 0) patrol_step++; break; // waiting time to
        arrive
    }
```

```

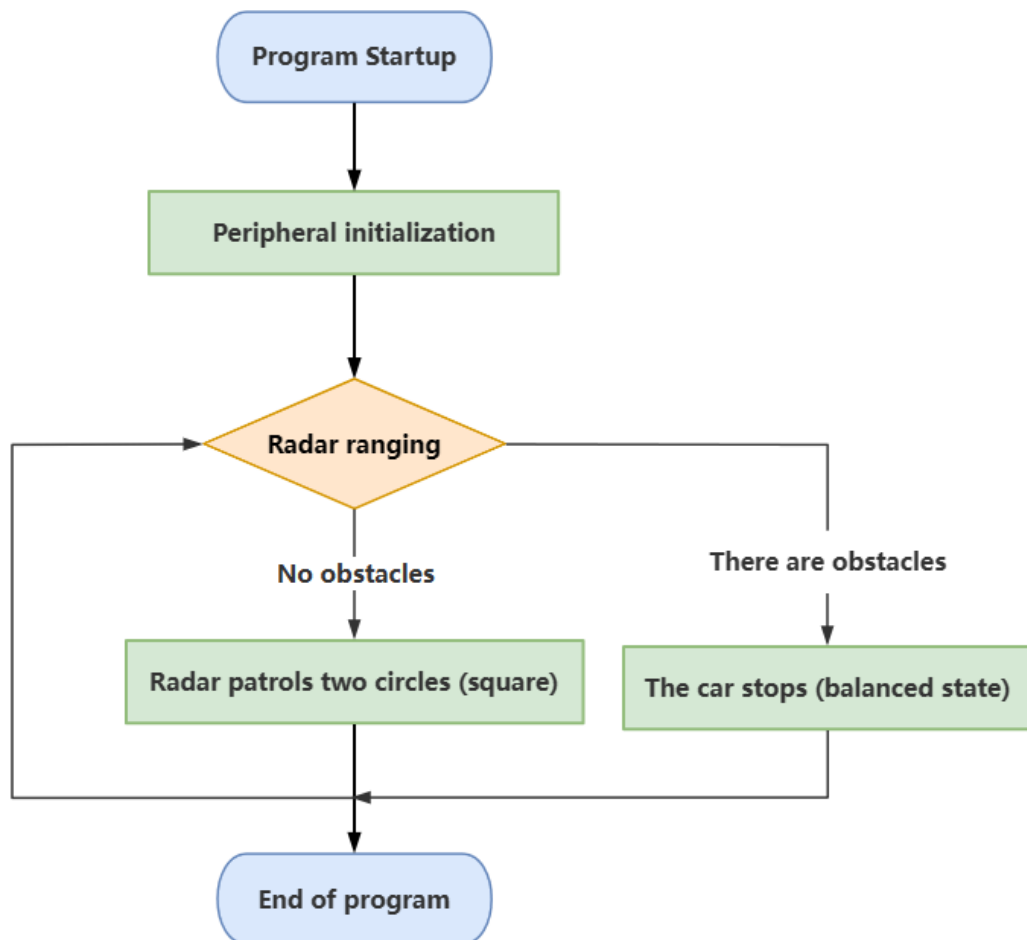
    case 2:myTurn_Kd = 35,Move_X = 0;Move_Z = 1350;
    delay_time_int(Trun_time);patrol_step++;break; // Turning
    case 3:if(get_time_int() == 0)patrol_step++;break; // waiting time to
arrive
    // Preventing jitter
    case 4:Move_X = 0;Move_Z = 0;delay_time_int(30);patrol_step++;break;//
stop
    case 5:if(get_time_int() == 0)patrol_step++;break; // waiting time to
arrive

    case 6:
        if(runtimes < (patrol_times*4) )
        {
            runtimes++;
            patrol_step = 0;
        }else patrol_step =7;// Reset button clear
        break;
    case 7:Move_X = 0;Move_Z = 0;// stop
    }
}

```

Program flow chart

Briefly introduce the process of function implementation:



Experimental phenomenon

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

Product supporting materials source code path: Attachment → Source code summary → 5.Balanced_Car_Extended → 15.Balance_Radar_Patrol

Experimental phenomenon

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

The balance car will patrol twice (the patrol track is a square). If an obstacle is encountered during the patrol, the patrol will stop and an alarm will be sounded (buzzer sounds); if the obstacle is removed, the alarm will be eliminated and the previously unfinished patrol will be resumed.

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

Common situations for triggering 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