1. Radar UART Output Protocol

UAV-R21-1 77GHz MMwave radar is UART output, 115200bps, 8N1.

Output Frequency 20Hz. Specific protocol format is as follows：

Header Byte D1 D2 D3 D4 D5 D6 D7 D8 D9 CRC

Byte	Parameter Description		Unit	Explanation	Remark
Byte 0	Header Byte 1	uint8_t		Fixed'T', that is 0x54	
Byte 1	Header Byte 2	uint8_t		Fixed'H', that is 0x48	
Byte 2~3	D1	uin16_t	cm	0 degree sector obstacle distance	
Byte4~5	D2	uin16_t	cm	45 degree sector obstacle distance	
Byte 6~7	D3	uin16_t	cm	90 degree sector obstacle distance	
Byte 8~9	D4	uin16_t	cm	135 degree sector obstacle distance	
Byte10~11	D5	uin16_t	cm	180 degree sector obstacle distance	
Byte 12~13	D6	uin16_t	cm	225 degree sector obstacle distance	
Byte 14~15	D7	uin16_t	cm	270 degree sector obstacle distance	
Byte 16~17	D8	uin16_t	cm	315 degree sector obstacle distance	
Byte 18	D9	Uint8_t		Fixed, 'M' ascii is decimal 77 (0x4D)	
Byte 19	CRC	Uin8_t		CRC8 check	See description below

Obstacle Distance Unit: millimeter. The high 8 bits are in front, the low 8 bits are in the back.

for example, 0 degree sector obstacle distance 0x07D0.

Byte 2=0x07, byte 3=0xD0, then the actual distance is 2m.

Note: It must be sent regardless of whether there is radar data or not. When the data is invalid, DX is filled with 0XFFFF. UAV-R22-1 77GHz MMwave radar obstacle avoidance system outputs D1, D2, D3, D4...D8 sector obstacle distances, and other sectors are invalid data. Fill with 0xFFFF.

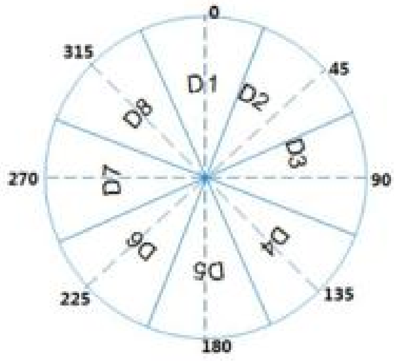

Figure 1 Sector Diagram

Radar output CRC8 check

Crc.cpp：

static const uint8_t crc8_table[] = {
0x00, 0x07, 0x0e, 0x09, 0x1c, 0x1b, 0x12, 0x15, 0x38, 0x3f, 0x36, 0x31, 0x24, 0x23, 0x2a, 0x2d, 0x70, 0x77, 0x7e, 0x79, 0x6c, 0x6b, 0x62, 0x65, 0x48, 0x4f, 0x46, 0x41, 0x54, 0x53, 0x5a, 0x5d, 0xe0, 0xe7, 0xee, 0xe9, 0xfc, 0xfb, 0xf2, 0xf5, 0xd8, 0xdf, 0xd6, 0xd1, 0xc4, 0xc3, 0xca, 0xcd, 0x90, 0x97, 0x9e, 0x99, 0x8c, 0x8b, 0x82, 0x85, 0xa8, 0xaf, 0xa6, 0xa1, 0xb4, 0xb3, 0xba, 0xbd, 0xc7, 0xc0, 0xc9, 0xce, 0xdb, 0xdc, 0xd5, 0xd2, 0xff, 0xf8, 0xf1, 0xf6, 0xe3, 0xe4, 0xed, 0xea, 0xb7, 0xb0, 0xb9, 0xbe, 0xab, 0xac, 0xa5, 0xa2, 0x8f, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9d, 0x9a, 0x27, 0x20, 0x29, 0x2e, 0x3b, 0x3c, 0x35, 0x32, 0x1f, 0x18, 0x11, 0x16, 0x03, 0x04, 0x0d, 0x0a, 0x57, 0x50, 0x59, 0x5e, 0x4b, 0x4c, 0x45, 0x42, 0x6f, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7d, 0x7a, 0x89, 0x8e, 0x87, 0x80, 0x95, 0x92, 0x9b, 0x9c, 0xb1, 0xb6, 0xbf, 0xb8, 0xad, 0xaa, 0xa3, 0xa4, 0xf9, 0xfe, 0xf7, 0xf0, 0xe5, 0xe2, 0xeb, 0xec, 0xc1, 0xc6, 0xcf, 0xc8, 0xdd, 0xda, 0xd3, 0xd4, 0x69, 0x6e, 0x67, 0x60, 0x75, 0x72, 0x7b, 0x7c, 0x51, 0x56, 0x5f, 0x58, 0x4d, 0x4a, 0x43, 0x44, 0x19, 0x1e, 0x17, 0x10, 0x05, 0x02, 0x0b, 0x0c, 0x21, 0x26, 0x2f, 0x28, 0x3d, 0x3a, 0x33, 0x34, 0x4e, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5c, 0x5b, 0x76, 0x71, 0x78, 0x7f, 0x6a, 0x6d, 0x64, 0x63, 0x3e, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2c, 0x2b, 0x06, 0x01, 0x08, 0x0f, 0x1a, 0x1d, 0x14, 0x13, 0xae, 0xa9, 0xa0, 0xa7, 0xb2, 0xb5, 0xbc, 0xbb, 0x96, 0x91, 0x98, 0x9f, 0x8a, 0x8d, 0x84, 0x83, 0xde, 0xd9, 0xd0, 0xd7, 0xc2, 0xc5, 0xcc, 0xcb, 0xe6, 0xe1, 0xe8, 0xef, 0xfa, 0xfd, 0xf4, 0xf3

```
} ;
uint8_t crc_crc8(const uint8_t *p, uint8_t len)
{
    uint16_t i; uint16_t crc = 0x0;
    while (len--)
    {
        i = (crc ^ *p++) & 0xFF; crc = (crc8_table[i] ^ (crc << 8)) & 0xFF;
    }
        return crc & 0xFF;
}
```

Debug way： crc8 = crc_crc8(buffer, 1); //buffer is the data receiving buffer array

2. System obstacle avoidance solution

In order to facilitate users to integrate UAV- R22-1 obstacle avoidance radar with open source flight control faster, Microbrain proposes following system solution, aiming to the current mainstream flight control platforms in the market.

Working out the compatible firmware of UAV-R22-1 for open source flight control obstacle avoidance protocol and normal open source flight control.

Integrated application settings of UAV-R22-1 on the APM flight control platform.

Flight control hardware： PixhawkV3/Orange Cube

Flight control software： ArduPilot Copter 4.3.8

Ground station software： MissionPlanner 1.3.81

2.1 Hardware installation and connection

2.1.1 Radar installation and wiring

The radar serial port is connected to any serial port of Telem1/Telem2/GPS2 of the flight control. Please note the radar needs to be powered separately. RX/TX are connected alternately, and the radar needs to be grounded. The radar interface is defined as shown below:

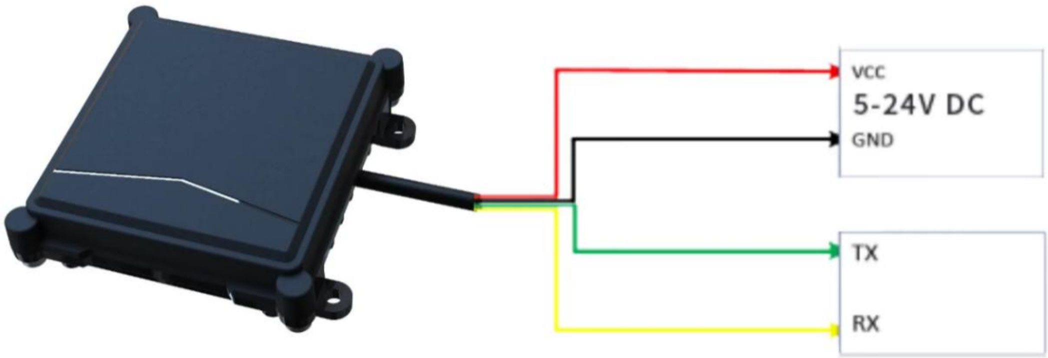

Figure 2 Radar interface Definition

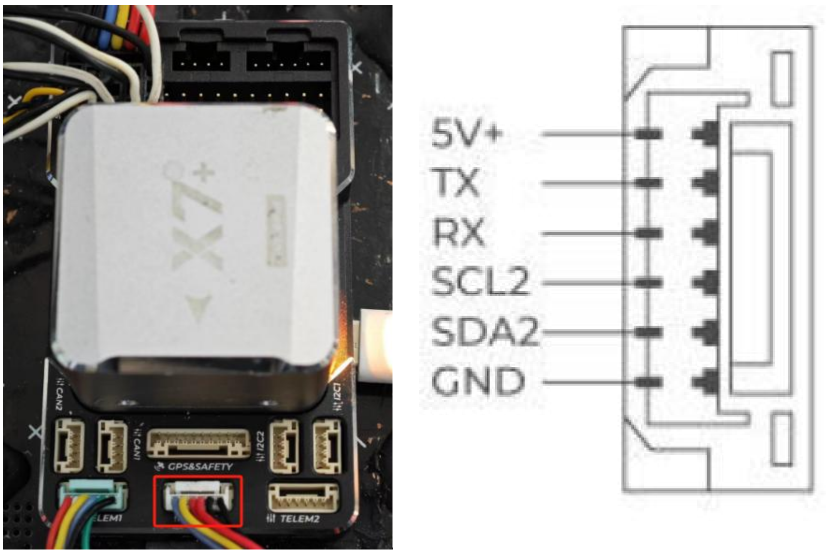

Figure 3 Flight control interface Definition

2.1.2 MissionPlanner Ground station flight control parameter settings

1. Open Mission Planner - CONFIG - FULL Parameter List

2. Connect the flight control UART per the radar's request, Corresponding search Serial1/Serial2/Serial4,（Take SERIAL4 as example, SERIAL4 Baudrate 115200bit/s（SERIAL4_BAUD set 115) The communication protocol set Lidar360（SERIAL2_PROTOCOL set 11), as shown below：

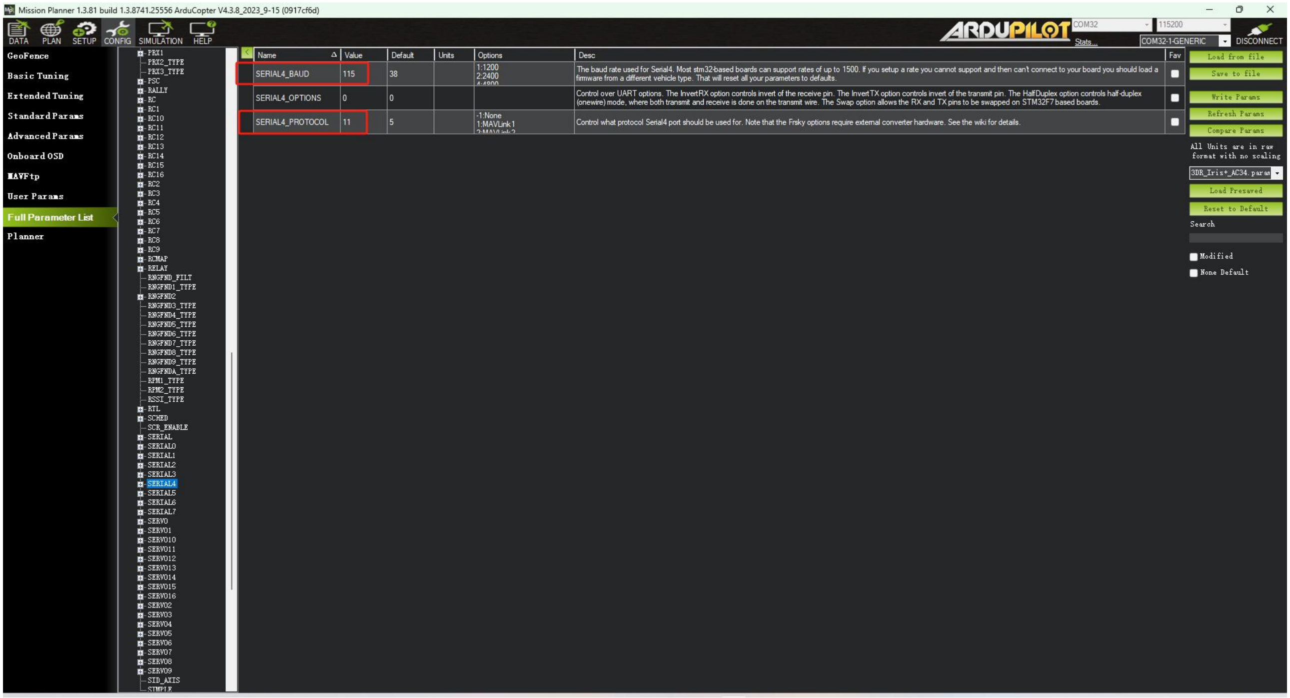

Figure 4 UART Setting

3. Set the obstacle avoidance radar protocol TeraRangerTowerEvo (PRX_TYPE1 set 6), as shown below：

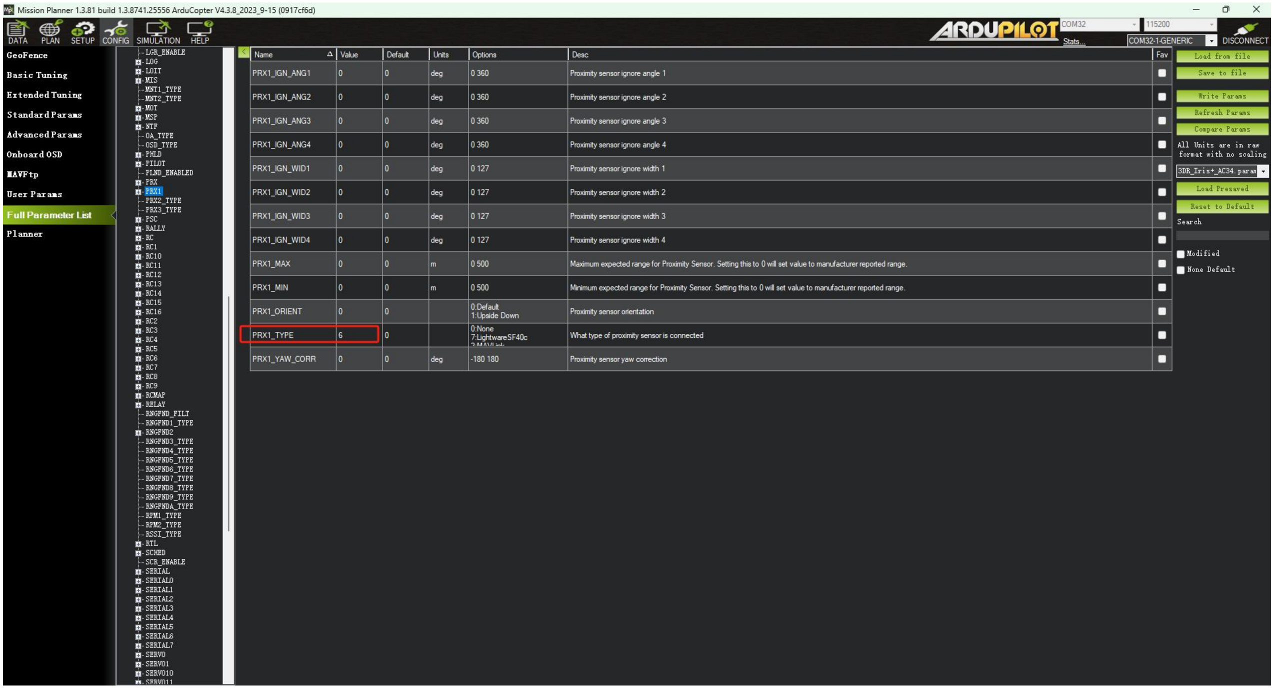

Figure 5　Obstacle avoidance radar protocol

4．Set AVOID_ENABLE as UseProximitySensor, Obstacle avoidance maximum tilt angle degree is 10 degrees(AVOID_ANGLE_MAX set 1000), Obstacle avoidance action mode set stop(AVOID_BEHAVE set 1). Under gps mode, the obstacle avoidance distance is 3m(AVOID_MARGIN set 3). Under fixed height, the obstacle avoidance distance is 10m(AVOID_DIST_MAX set 10), as shown below：

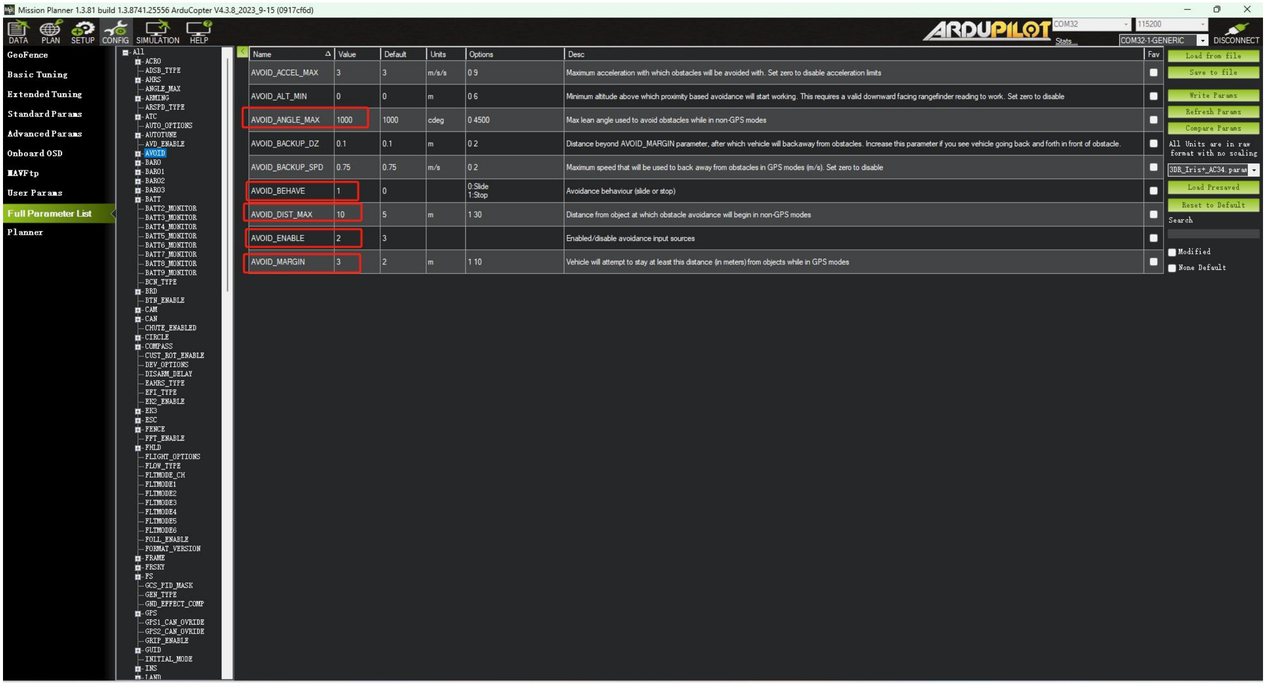

Figure 6 Set obstacle avoidance parameters

5. Write Params after setting the above parameters, restart the flight control, connect to MissionPlanner, CTRL+F to open debugging window to see the obstacle avoidance distance display as below pic(The target is 2.6m in sector 0).

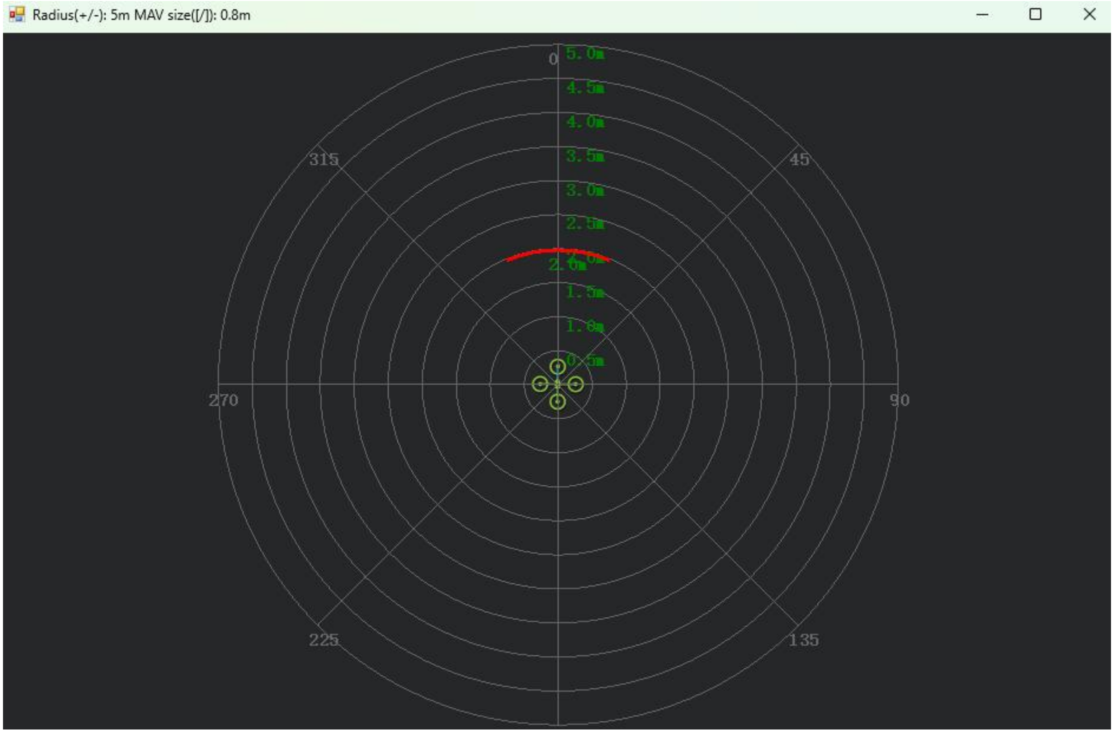

Figure7 Obstacle avoidance distance demonstration