77GHz Millimeter-wave Obstacle Avoidance Radar

User Manual

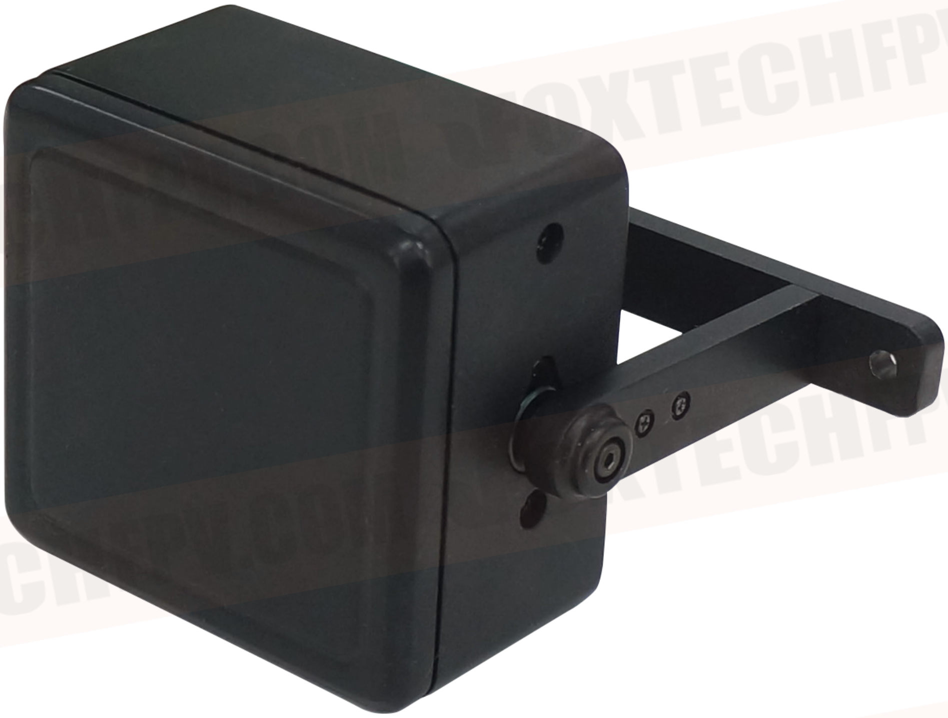
V1.0 2019.06

FOXTECH

Contents

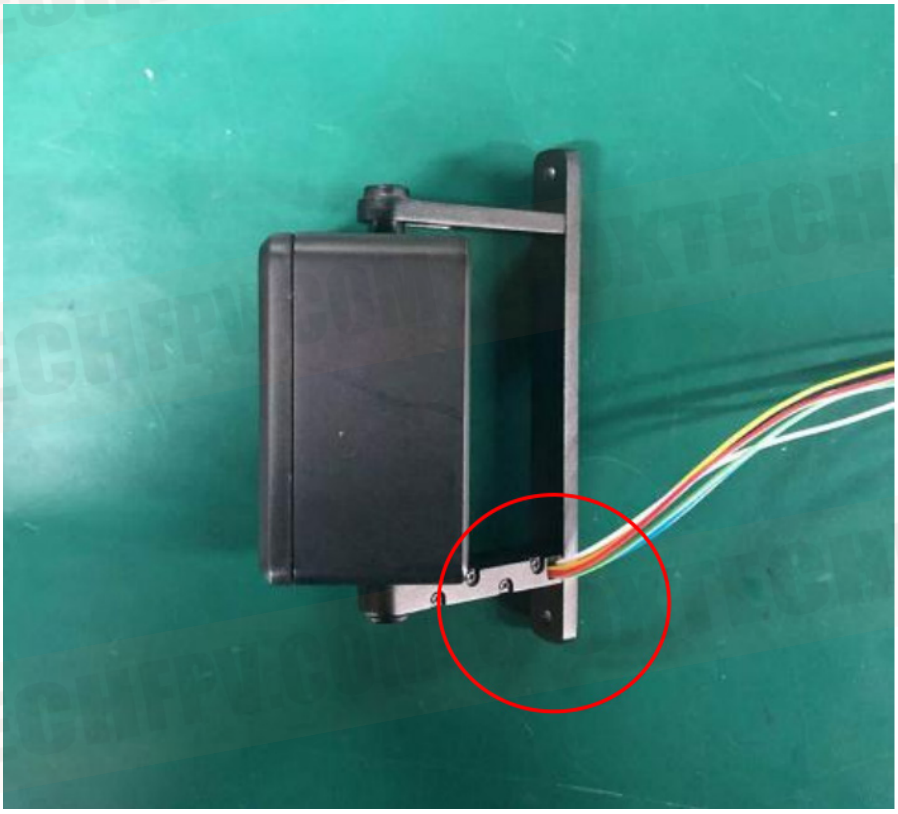
Overview

The 77GHz millimeter-wave obstacle avoidance radar is especially designed for industrial UAV.
The obstacle avoidance radar can be easily integrated with existing platforms, like PIXHAWK, DJI A3,N3 etc.
It is a single axis obstacle avoidance radar system, which is able to maintain a self-stabilizing state during flight, and always detecting forward, can not be affected by the flight action of the drone.



Note

(1) The power supply pin needs to be externally connected to 5VDC;
(2) The front of the module is consistent with the direction of the drone head when installing, and there is no obstruction in front of the module;
(3) Pay attension to the installation mode of the gimbal,the outgoing line side is the front side.



Specification

Transmission Frequency	76-77GHZ
Detection Range	Greater Than 100m
Detection Accuracy	± 0.18m
Wave Speed Width	110°(Yaw) and 15.6°(pitch)
Interface	UART
Data Output Freauency	50Hz
Voltage	5V(DC)
Gimbal Angle Range	180°
Working Temperature	-20℃
Weight	120g
Size	70x102x34mm

Pin Interface Definition

Pin Interface Definition		
VCC(Red)	5V DC	-
GND(Black)	-	-
UART_RX(Yellow)	TTL 3.3 DC	Radar Port
UART_TX(Green)	TTL 3.3 DC	Radar Port
UART_RX(White)	TTL 3.3 DC	Gimbal Port
UART_TX(Blue)	TTL 3.3 DC	Gimbal Port

Adjust the self-stabilizing angle through the serial port.

Single Module Test

Preparation Before Test

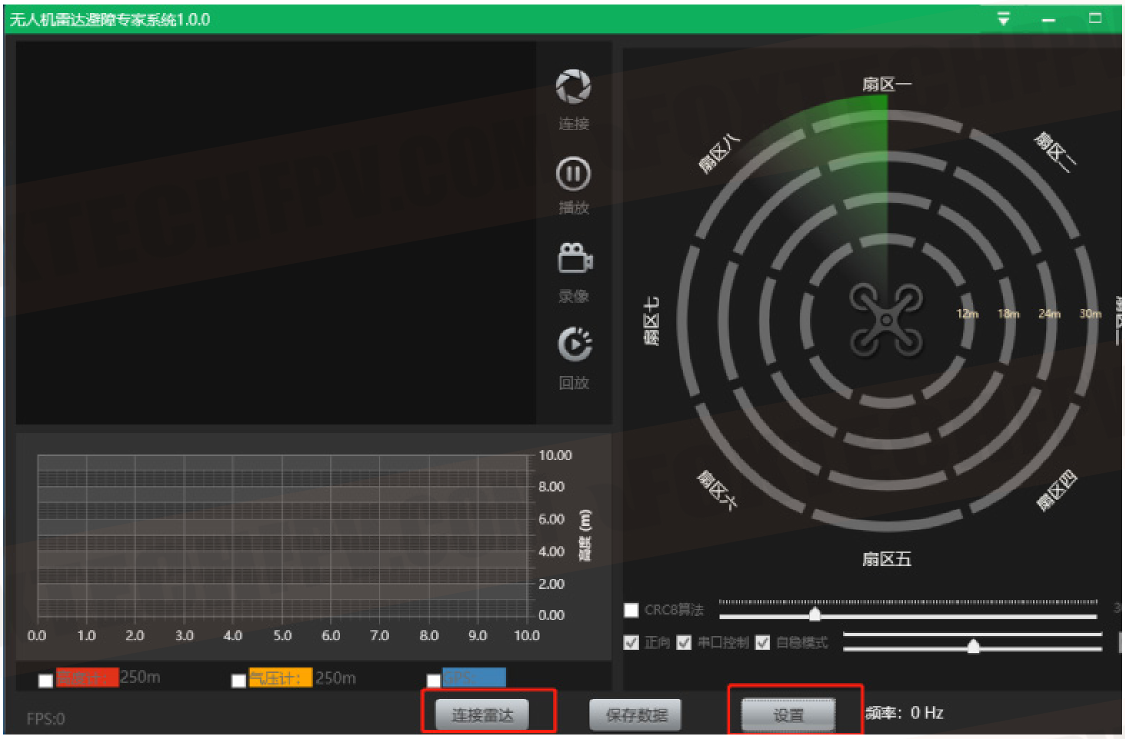
Use USB to serial port device to connect the radar output serial port, the USB port is connected to the PC serial port assistant debugging software, you can see the output data, or observe the radar output data more intuitively through the company's developed software “UAV radar obstacle avoidance expert system”. Please refer to the serial data protocol description for specific data. The test using tools or software is shown in the following table:

No.	Equipent	Quant.
1	77G Radar	1
2	PC	1
3	USB to TTL Adapter	1
4	5V Power Adapter	1
5	Serial port debugging software/radar obstacle avoidance expert system.	1

Please connect the TX of the adapter to the RX of the radar, and the RX to the TX.

Connect to Radar

Connect the radar serial port to PC according to the connecting method above, open the "radar obstacle avoidance expert system", and you will see the radar output data in the sector aera the interface is as below:



无人机雷达避障专家系统1.0.0

□ CRC8算法

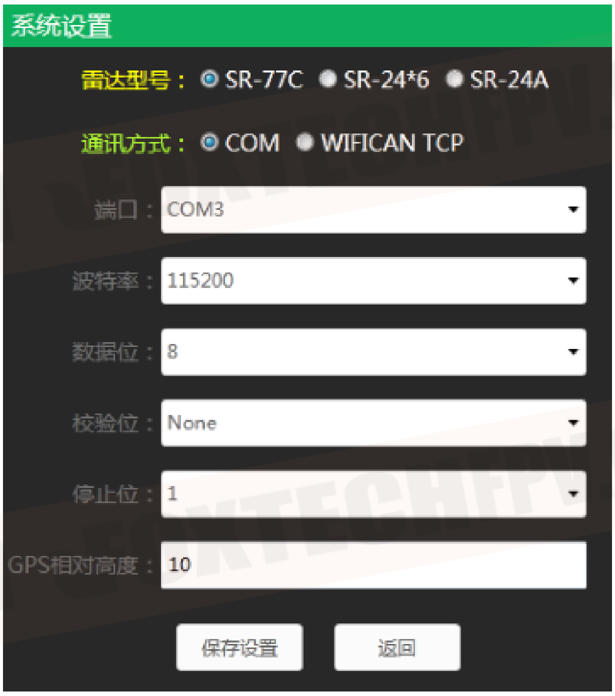
☑ 正向 ☑ 串口控制 ☑ 自稳模式

□ 高度计：250m □ 气压计：250m □ GPS:

FPS:0

频率：0 Hz

The default connecting method of this software is COM.After connect the radar to PC, open the software and it will automatically identify the port, chose the radar type, click "save the setting " .



系统设置

雷达型号： ◉ SR-77C ○ SR-24*6 ○ SR-24A

通讯方式： ◉ COM ○ WIFICAN TCP

端口： COM3

波特率： 115200

数据位： 8

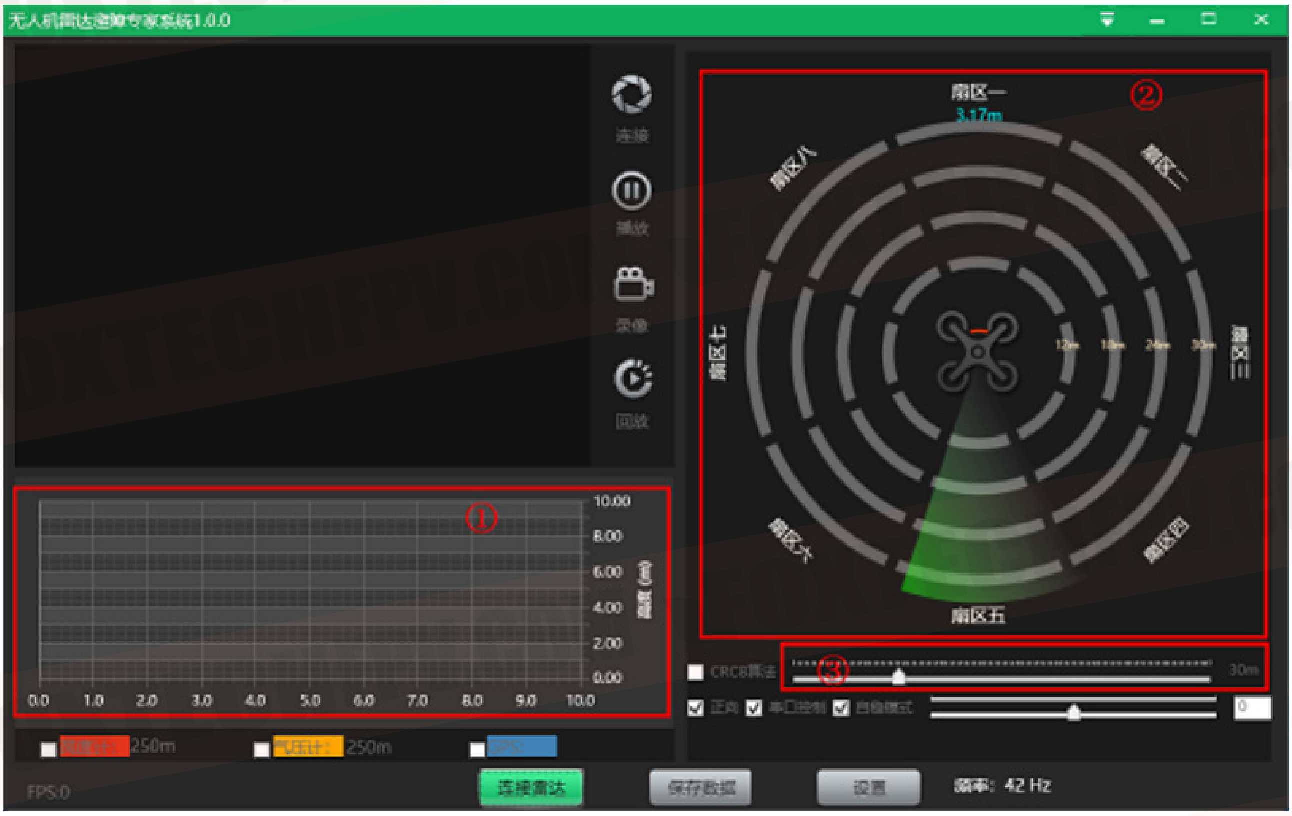
校验位： None

停止位： 1

GPS相对高度： 10

Data Real-time Display

The real-time data display module is based on the successful connection of the device, and displays the real-time received radar data in dynamic line graph and sector orientation:



无人机雷达避障专家系统1.0.0

频率: 42 Hz

FPS:0

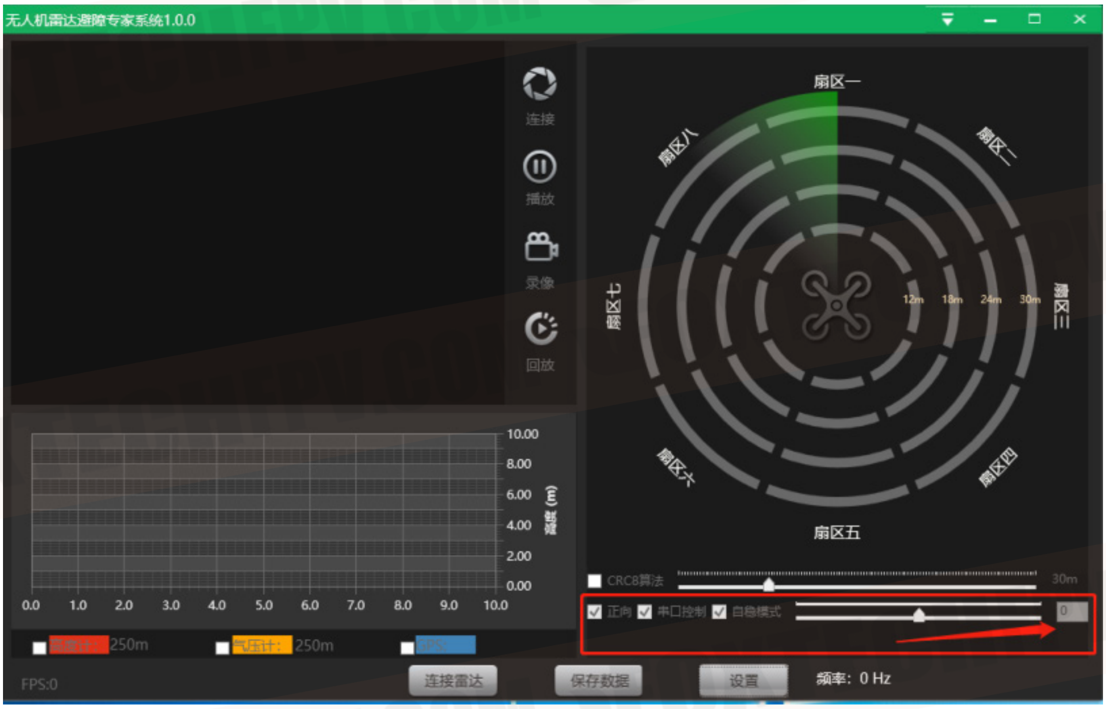
-In the above figure 1, the radar altimeter, barometer and other data are displayed in real time in a dynamic line diagram;

-In the above figure 2, the sector data of the obstacle avoidance radar is displayed in real time in the sector orientation map, and the obstacle avoidance display range can be dynamically adjusted in the above figure 3, the range is 0-120m, the default is 30m, and according to the return The obstacle distance is displayed in an arc of a different color in the corresponding sector, and the distance of the obstacle is displayed under the corresponding sector;

-According to the data algorithm returned by the radar, you can freely check the CRC8 algorithm on the left side of Figure 3. The default is not checked. After checking, the check value will be calculated by the custom CRC8 algorithm, which is mainly used for parameter setting.

Gimbal Control Module

The gimbal control module is based on the successfull connection of the radar. It is necessary to connect the giambal serial port to the PC and control the gimbal to achieve stable radar obstacle avoidance reliability. The operation commands are as shown in the red box below:



☑ 正向 ☑ 串口控制 ☑ 自稳模式 0

-For the forward and reverse of the actual installation of the gimbal,the default is to check the “Forward” checkbox, that is, the gimbal installation is its own forward direction, otherwise the installation is reversed, then the “Forward” checkbox is unchecked;

-The gimbal control mode is divided into fixed mode and self-stabilized mode. In each mode, there are parameter setting and serial port control. The system can read the current gimbal control mode in real time, and can be freely set according to requirements.

-The system can read the gimbal angle in real time. This angle distinguishes different modes. The fixed mode is the gimbal fixed angle,the self-stabilizing mode is the gimbal stable angle. The angle range is -90 degrees to 90 degrees. At the same time, you can freely drag the arrow position progress bar or change the text value to change the angle.

The Radar has default parameters, which generally do not need to be changed. If necessary, it can be modified according to the 77G radar head protocol.

Data Protocol

The 77G radar is a dual serial port output, one outputs radar data and the other is a gimbal output serial port. The serial port settings are 115200bps, 8N1.

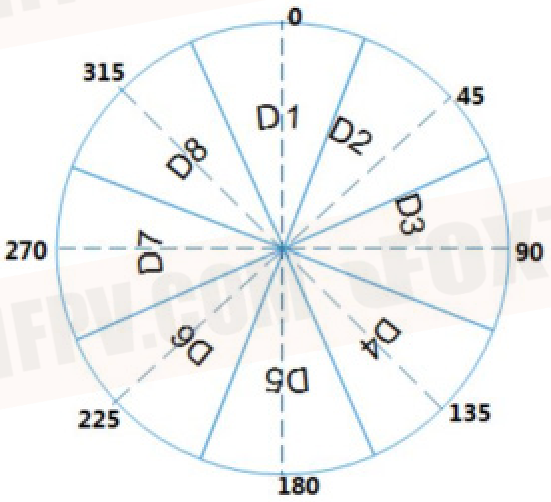
Radar Serial Port Output Protocol

77G Radar serial port output frequency is 50Hz, the specific protocol format is as follows:
head byte D1 D2 D3 D4 D5 D6 D7 D8 CRC8

Byte	Parametric Description	Type	Unit	Explanation	Note
Byte 0	Lead Byte 1	uint8_t		Fixed as' T ', namely 0x54	
Byte 1	Lead Byte 2	uint8_t		Fixed as' H ', namely 0x48	
Byte 2~3	D1	uint16_t	cm	0 degree sector obstacle distance	
Byte 4~5	D2	uint16_t	cm	45 degree sector obstacle distance	
Byte 6~7	D3	uint16_t	cm	90 degree sector obstacle distance	
Byte 8~9	D4	uint16_t	cm	0135 degree sector obstacle distance	
Byte 10~11	D5	uint16_t	cm	180 degree sector obstacle distance	
Byte 12~13	D6	uint16_t	cm	225 degree sector obstacle distance	
Byte 14~15	D7	uint16_t	cm	270 degree sector obstacle distance	
Byte 016~17	D8	uint16_t	cm	315 degree sector obstacle distance	
Byte 18	CRC8	uint8_t		CRC8 verification	See below description

Obstacle distance: unit: cm; The high 8 bits are in front and the low 8 bits are behind, such as 0-degree sector obstacles distance 0x07D0, byte 2=0x07, byte 3=0xD0, the actual distance is 20m.

Send data no matter if it has radar data or not. When the data is invalid, DX fills in 0xFFFF. 77G radar obstacle avoidance system output D1, D2, D8 sector obstacle distance, other sectors are invalid data, filled with 0xFFFF.



Radar Output CRC8 Calibration

```
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;
}
```

77G Radar Gimbal Protocol

Configuring Gimbal Parameters

gimbal parameter protocol setting by the PC-side serial port is shown in the following table. A single instruction has no return value, and the status is continuously sent by the gimbal.

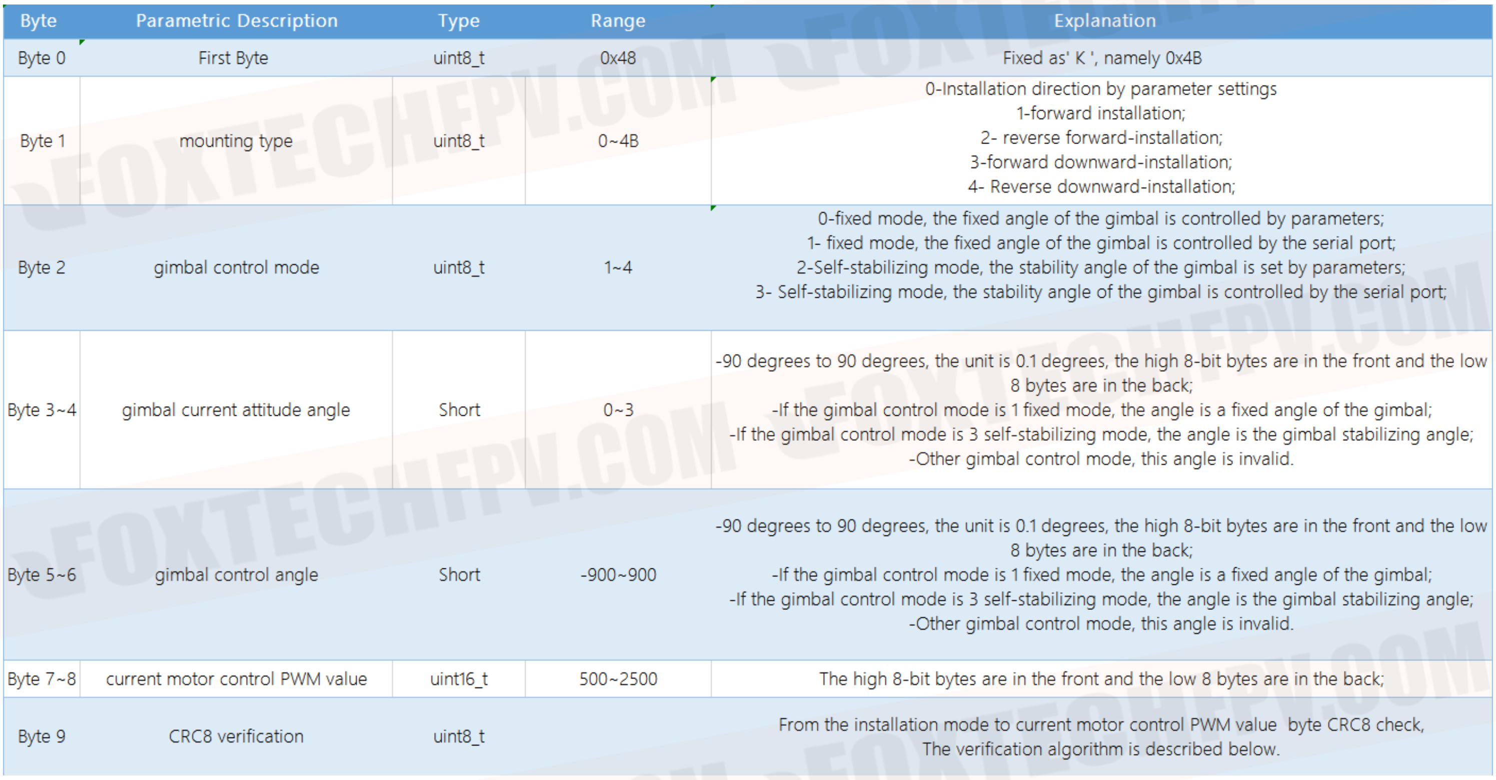
Byte	Parametric Description	Type	Range	Explanation
Byte 0	First Byte	uint8_t	0x48	Fixed as' K ', namely 0x4B

Byte 1	mounting type	uint8_t	0~4, default 0x00	0-Installation direction by parameter settings
				1-forward installation;
				2- reverse forward-installation;
				3-forward downward-installation;
				4- Reverse downward-installation;

Byte 2	gimbal control mode	uint8_t	0~3, default 0x02	0-fixed mode, the fixed angle of the gimbal is controlled by parameters; 1- fixed mode, the fixed angle of the gimbal is controlled by the serial port; 2-Self-stabilizing mode, the stability angle of the gimbal is set by parameters; 3- Self-stabilizing mode, the stability angle of the gimbal is controlled by the serial port;
Byte 3~4	gimbal angle	Short	-900~900, default 0	-90 degrees to 90 degrees, the unit is 0.1 degrees, the upper 8 bytes are in the front and the lower 8 bytes are in the back; -If the gimbal control mode is 1 fixed mode, the angle is a fixed angle of the gimbal; -If the gimbal control mode is 3 self-stabilizing mode, the angle is the gimbal stabilizing angle; -Other gimbal control mode, this angle is invalid
Byte 5	CRC8 verification	uint8_t		From the installation mode to the gimbal stable angle byte CRC8 check, The verification algorithm is described below.

Gimbal Delivery Status

The gimbal delivery status protocol is as shown in the following table. The output frequency is 100Hz.



Byte	Parametric Description	Type	Range	Explanation
Byte 0	First Byte	uint8_t	0x48	Fixed as' K ', namely 0x4B

Byte 1	mounting type	uint8_t	0~4B	0-Installation direction by parameter settings
				1-forward installation;
				2- reverse forward-installation;
				3-forward downward-installation;
				4- Reverse downward-installation;

Byte 2	gimbal control mode	uint8_t	1~4	0-fixed mode, the fixed angle of the gimbal is controlled by parameters; 1- fixed mode, the fixed angle of the gimbal is controlled by the serial port; 2-Self-stabilizing mode, the stability angle of the gimbal is set by parameters; 3- Self-stabilizing mode, the stability angle of the gimbal is controlled by the serial port;
Byte 3~4	gimbal current attitude angle	Short	0~3	-90 degrees to 90 degrees, the unit is 0.1 degrees, the high 8-bit bytes are in the front and the low 8 bytes are in the back; -If the gimbal control mode is 1 fixed mode, the angle is a fixed angle of the gimbal; -If the gimbal control mode is 3 self-stabilizing mode, the angle is the gimbal stabilizing angle; -Other gimbal control mode, this angle is invalid.
Byte 5~6	gimbal control angle	Short	-900~900	-90 degrees to 90 degrees, the unit is 0.1 degrees, the high 8-bit bytes are in the front and the low 8 bytes are in the back; -If the gimbal control mode is 1 fixed mode, the angle is a fixed angle of the gimbal; -If the gimbal control mode is 3 self-stabilizing mode, the angle is the gimbal stabilizing angle; -Other gimbal control mode, this angle is invalid.
Byte 7~8	current motor control PWM value	uint16_t	500~2500	The high 8-bit bytes are in the front and the low 8 bytes are in the back;
Byte 9	CRC8 verification	uint8_t		From the installation mode to current motor control PWM value byte CRC8 check, The verification algorithm is described below.

PTZ CRC8 Check

PTZ CRC8 check procedure.

```
Crc.cpp
static const uint8_t crc8_table[] = {
0x39, 0x61, 0x58, 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,
```

SR-PA77A

27 B7

400 100 9892 www.sky-robotics.cn

```
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
};
/*
crc8 from trone driver by Luis Rodrigues
*/
uint8_t crc_crc8(const uint8_t *p, uint8_t len)
{
uint16_t i;
uint16_t crc = 0x0030;
while (len--) {
i = (crc ^ *p++) & 0xFF;
crc = (crc8_table[i] ^ (crc << 8)) & 0xFF;
}
return crc & 0xFF;
}
```

System Obstacle Avoidance Solution

In order to facilitate the faster and better integrated use with the radar module, we have proposed the following system solutions for the mainstream flight control platforms currently on the market.

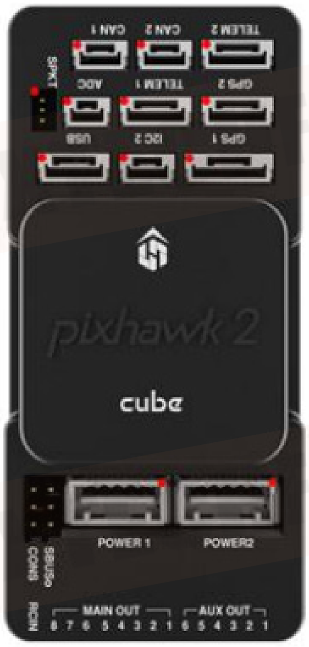
Open Source Flight Control Platform

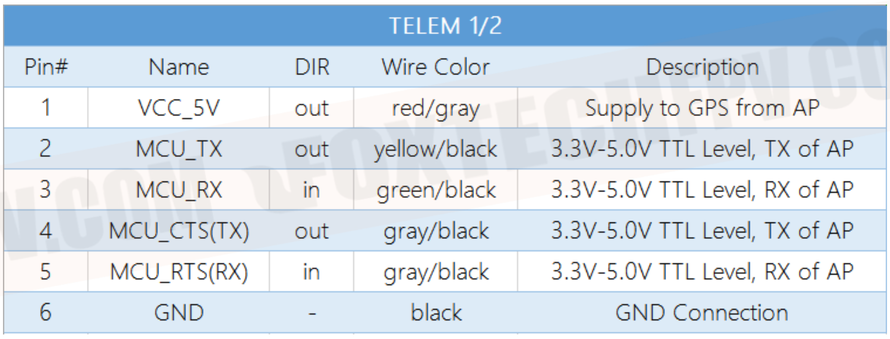
The radar is compatible with the open source flight control obstacle avoidance protocol and can be directly connected to the general open source flight control platform. The following is a brief description of the integrated application settings of this radar on the APM flight control platform.

Flight Control Hardware: PixhawkV3
Flight Control Software: ArduPilot Copter 3.5.5
Ground station software: MissionPlanner 1.3.62

Radar Installation and Wiring

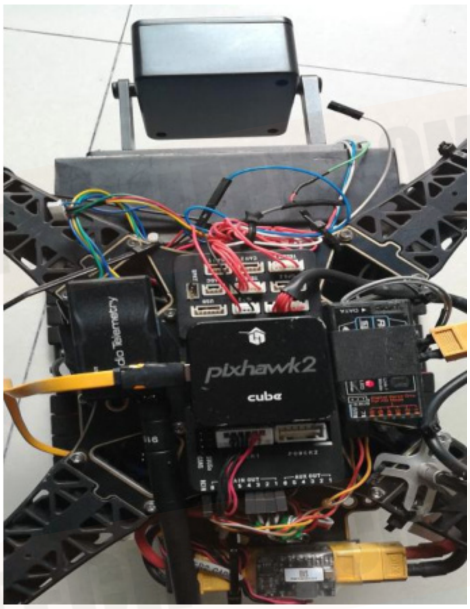
The radar serial port is connected to the pixhawk TELEM2 interface. The radar needs to be powered separately. The interface definition is as shown below:





TELEM 1/2				
Pin#	Name	DIR	Wire Color	Description
1	VCC_5V	out	red/gray	Supply to GPS from AP
2	MCU_TX	out	yellow/black	3.3V-5.0V TTL Level, TX of AP
3	MCU_RX	in	green/black	3.3V-5.0V TTL Level, RX of AP
4	MCU_CTS(TX)	out	gray/black	3.3V-5.0V TTL Level, TX of AP
5	MCU_RTS(RX)	in	gray/black	3.3V-5.0V TTL Level, RX of AP
6	GND	-	black	GND Connection

The installation method can refer to the following figure:

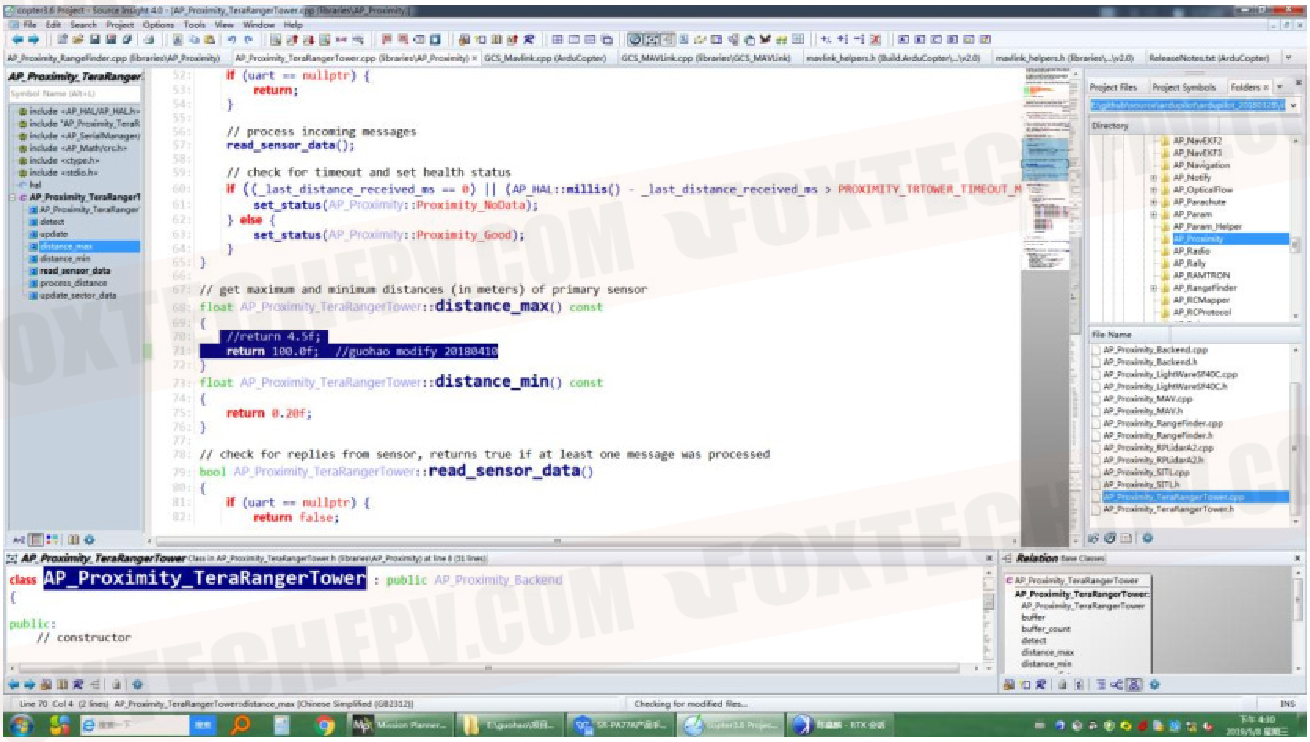


Flight Control Source Code Modification

The obstacle avoidance radar uses the Lidar360 protocol. Because the maximum measurement distance is limited, you need to modify the "AP_Proximity_TeraRangerTower.cpp" file in the AP_Proximity library in the libraries directory.

As shown below:

(1) Modify the maximum measurement distance to 100m

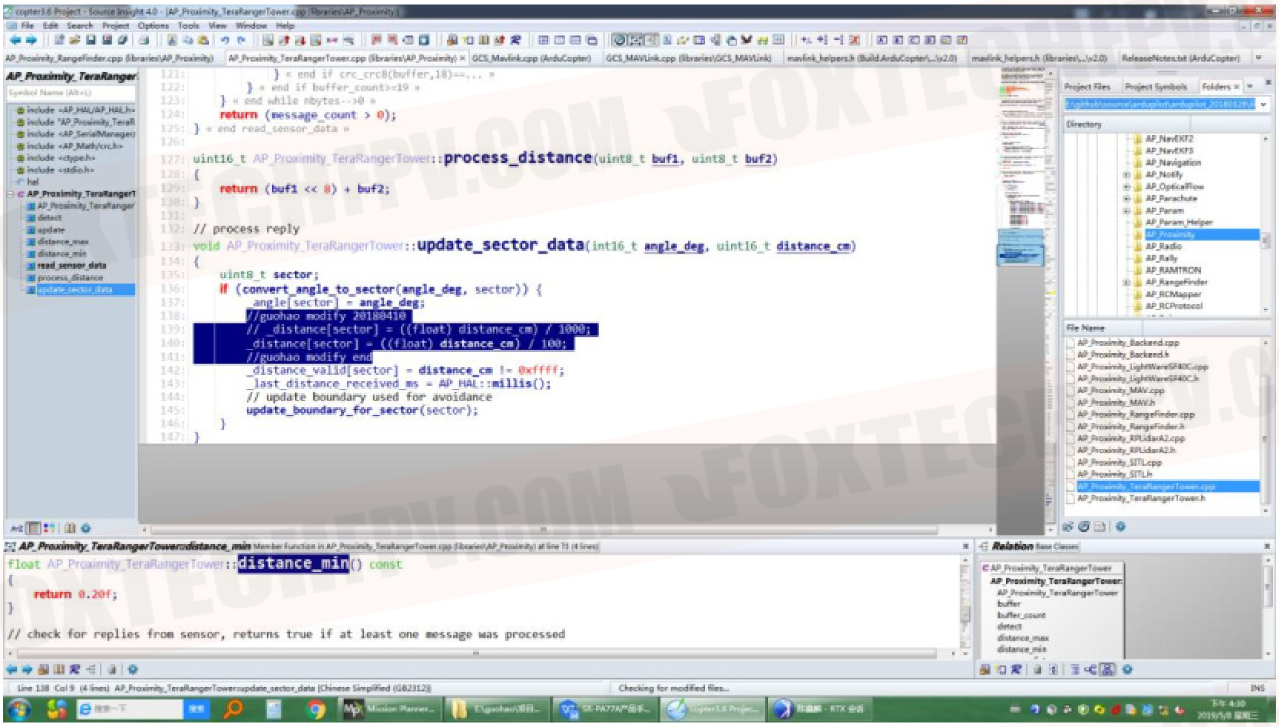


```
    if (uart == nullptr) {
        return;
    }

    // process incoming messages
    read_sensor_data();

    // check for timeout and set health status
    if ((_last_distance_received_ms == 0) || (AP_HAL::millis() - _last_distance_received_ms > PROXIMITY_TRTOWER_TIMEOUT_M
        set_status(AP_Proximity::Proximity_NoData);
    } else {
        set_status(AP_Proximity::Proximity_Good);
    }
}

// get maximum and minimum distances (in meters) of primary sensor
float AP_Proximity_TeraRangerTower::distance_max() const
{
    //return 4.5f;
    return 100.0f;  //guohao modify 20180410
}
float AP_Proximity_TeraRangerTower::distance_min() const
{
    return 0.20f;
}

// check for replies from sensor, returns true if at least one message was processed
bool AP_Proximity_TeraRangerTower::read_sensor_data()
{
    if (uart == nullptr) {
        return false;
```

```
class AP_Proximity_TeraRangerTower : public AP_Proximity_Backend
{

public:
    // constructor
```

Line 70 Col 4 (2 lines) AP_Proximity_TeraRangerTower::distance_max [Chinese Simplified (GB2312)]

(2) Modify the measured data unit to cm.



```
            } « end if crc_crc8(buffer,18)==... »
        } « end if buffer_count>=19 »
    } « end while nbytes-->0 »
    return (message_count > 0);
} « end read_sensor_data »

uint16_t AP_Proximity_TeraRangerTower::process_distance(uint8_t buf1, uint8_t buf2)
{
    return (buf1 << 8) + buf2;
}

// process reply
void AP_Proximity_TeraRangerTower::update_sector_data(int16_t angle_deg, uint16_t distance_cm)
{
    uint8_t sector;
    if (convert_angle_to_sector(angle_deg, sector)) {
        _angle[sector] = angle_deg;
        //guohao modify 20180410
        // _distance[sector] = ((float) distance_cm) / 1000;
        _distance[sector] = ((float) distance_cm) / 100;
        //guohao modify end
        _distance_valid[sector] = distance_cm != 0xffff;
        _last_distance_received_ms = AP_HAL::millis();
        // update boundary used for avoidance
        update_boundary_for_sector(sector);
    }
}
```

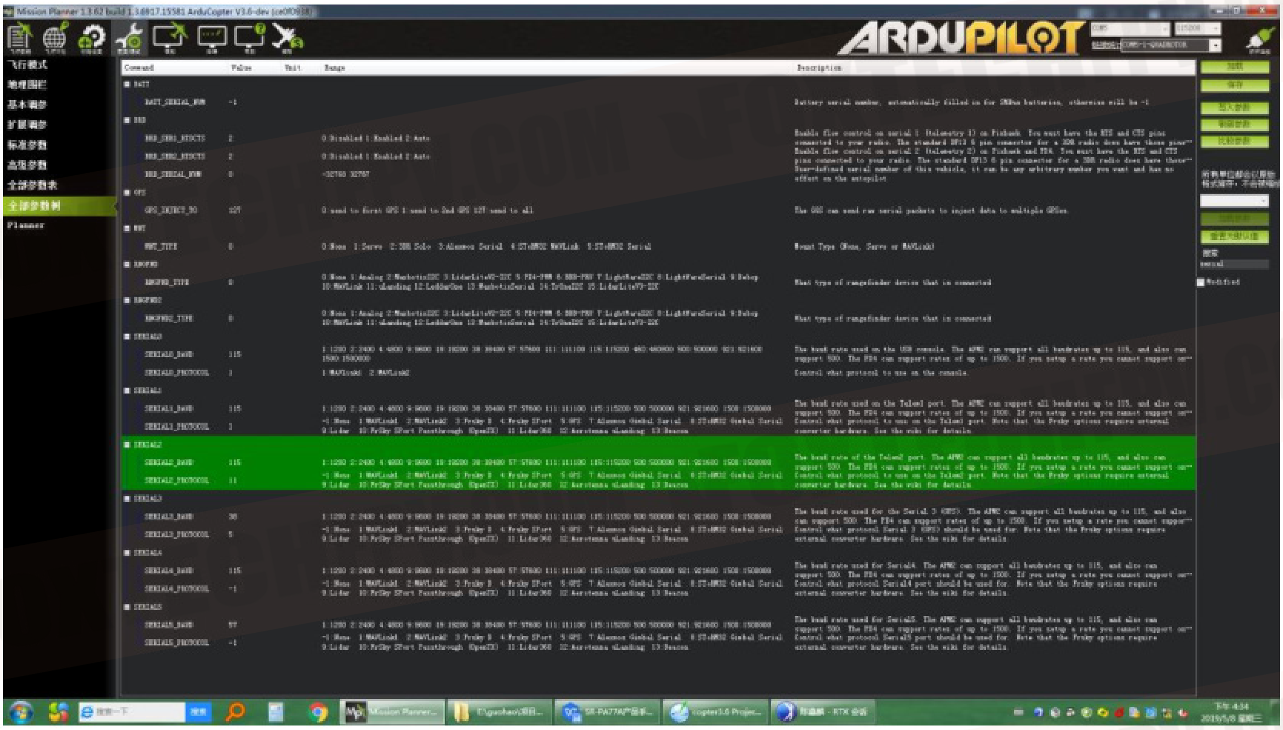
AP_Proximity_TeraRangerTower::distance_min Member Function in AP_Proximity_TeraRangerTower.cpp (libraries\AP_Proximity) at line 73 (4 lines)

```
float AP_Proximity_TeraRangerTower::distance_min() const
{
    return 0.20f;
}

// check for replies from sensor, returns true if at least one message was processed
```

Line 138 Col 9 (4 lines) AP_Proximity_TeraRangerTower::update_sector_data [Chinese Simplified (GB2312)]   Checking for modified files...

MissionPlanner Ground Station Flight Control Parameter Settings

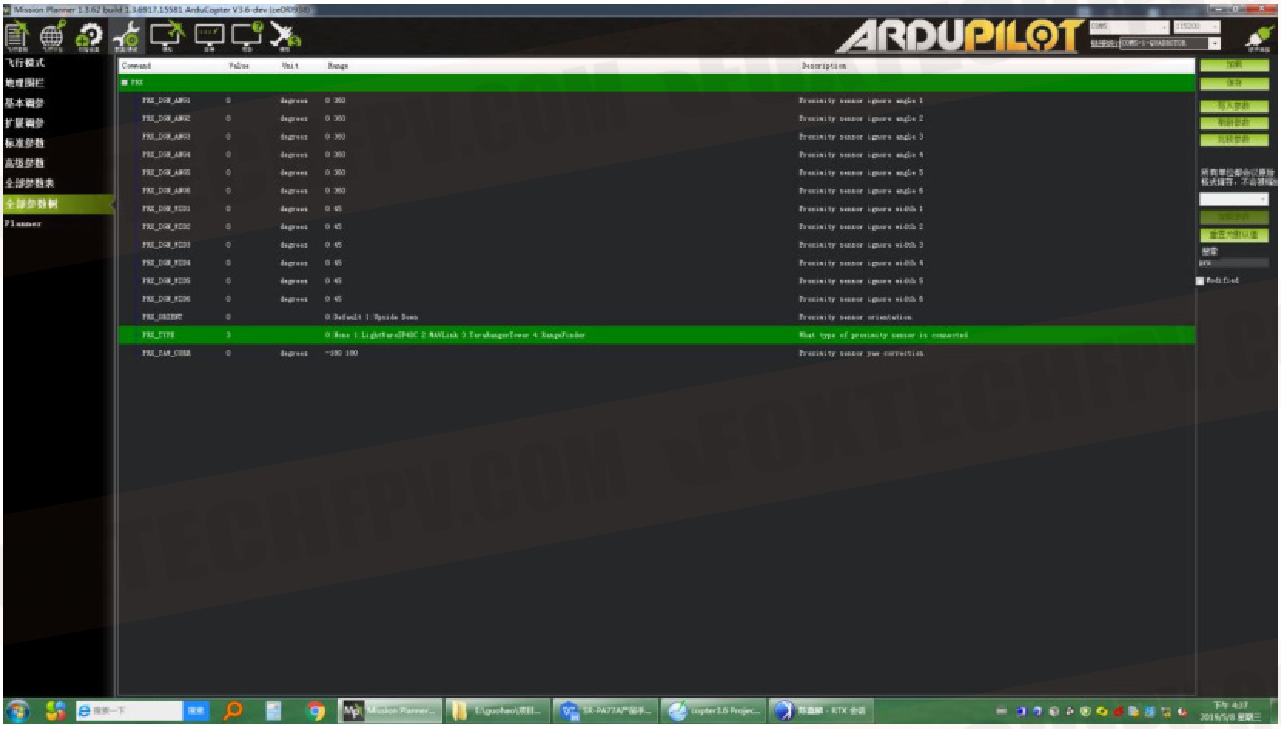
(1) Set the TELEM2 serial port parameter, the SERIAL2 baud rate is set to 115200bit/s (SERIAL2_BAUD is set to 115) and the communication protocol is set to Lidar360 (SERIAL2_PROTOCOL is set to 11) as shown below:



Mission Planner 1.3.62 build 1.3.6917.15581 ArduCopter V3.6-dev (ce0f0938)

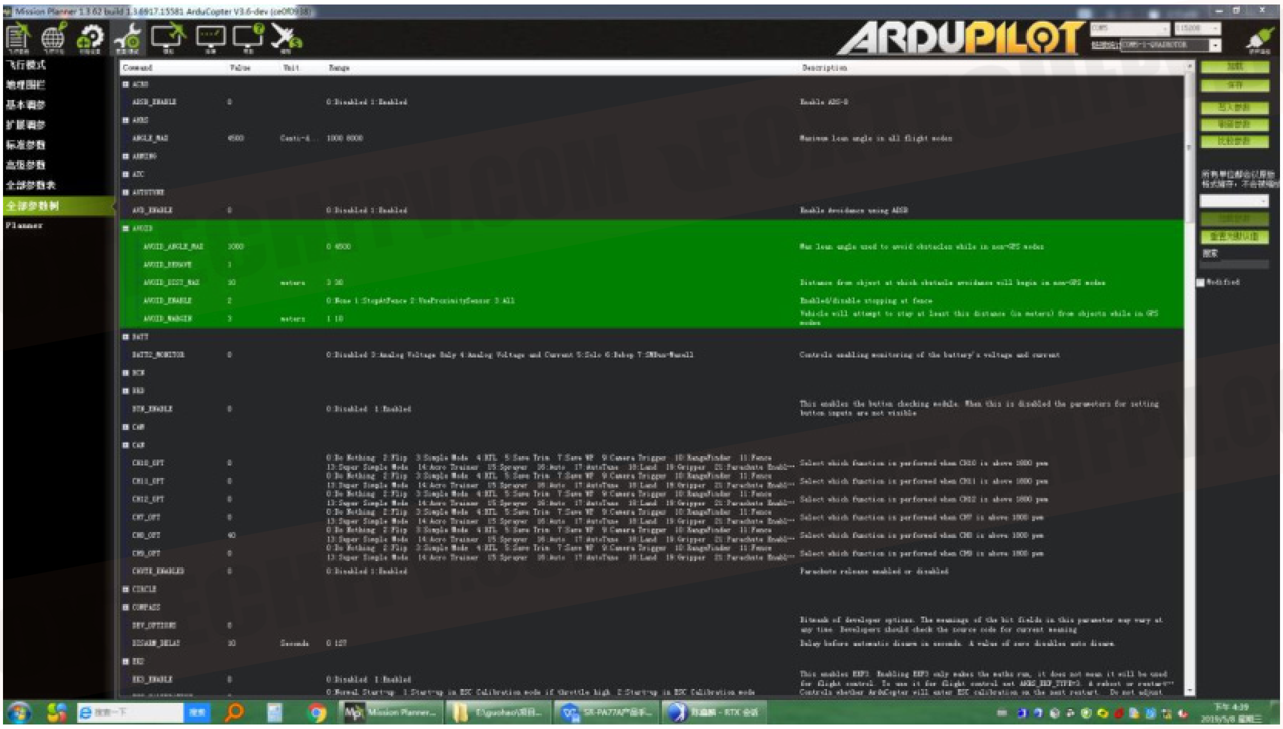
Command	Value	Unit	Range	Description
BATT				
BATT_SERIAL_NUM	-1			Battery serial number, automatically filled in for SMBus batteries, otherwise will be -1
BRD				
BRD_SER1_RTSCTS	2		0:Disabled 1:Enabled 2:Auto	Enable flow control on serial 1 (telemetry 1) on Pixhawk. You must have the RTS and CTS pins connected to your radio. The standard DF13 6 pin connector for a 3DR radio does have those pins…
BRD_SER2_RTSCTS	2		0:Disabled 1:Enabled 2:Auto	Enable flow control on serial 2 (telemetry 2) on Pixhawk and PX4. You must have the RTS and CTS pins connected to your radio. The standard DF13 6 pin connector for a 3DR radio does have those…
BRD_SERIAL_NUM	0		-32768 32767	User-defined serial number of this vehicle, it can be any arbitrary number you want and has no effect on the autopilot
GPS				
GPS_INJECT_TO	127		0:send to first GPS 1:send to 2nd GPS 127:send to all	The GGS can send raw serial packets to inject data to multiple GPSes.
MNT				
MNT_TYPE	0		0:None 1:Servo 2:3DR Solo 3:Alexmos Serial 4:STorM32 MAVLink 5:STorM32 Serial	Mount Type (None, Servo or MAVLink)
RNGFND				
RNGFND_TYPE	0		0:None 1:Analog 2:MaxbotixI2C 3:LidarLiteV2-I2C 5:PX4-PWM 6:BBB-PRU 7:LightWareI2C 8:LightWareSerial 9:Bebop 10:MAVLink 11:uLanding 12:LeddarOne 13:MaxbotixSerial 14:TrOneI2C 15:LidarLiteV3-I2C	What type of rangefinder device that is connected
RNGFND2				
RNGFND2_TYPE	0		0:None 1:Analog 2:MaxbotixI2C 3:LidarLiteV2-I2C 5:PX4-PWM 6:BBB-PRU 7:LightWareI2C 8:LightWareSerial 9:Bebop 10:MAVLink 11:uLanding 12:LeddarOne 13:MaxbotixSerial 14:TrOneI2C 15:LidarLiteV3-I2C	What type of rangefinder device that is connected
SERIAL0				
SERIAL0_BAUD	115		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 460:460800 500:500000 921:921600 1500:1500000	The baud rate used on the USB console. The APM2 can support all baudrates up to 115, and also can support 500. The PX4 can support rates of up to 1500. If you setup a rate you cannot support or…
SERIAL0_PROTOCOL	1		1:MAVLink1 2:MAVLink2	Control what protocol to use on the console.
SERIAL1				
SERIAL1_BAUD	115		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 500:500000 921:921600 1500:1500000	The baud rate used on the Telem1 port. The APM2 can support all baudrates up to 115, and also can support 500. The PX4 can support rates of up to 1500. If you setup a rate you cannot support or…
SERIAL1_PROTOCOL	1		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Lidar 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 12:Aerotenna uLanding 13:Beacon	Control what protocol to use on the Telem1 port. Note that the Frsky options require external converter hardware. See the wiki for details.
SERIAL2				
SERIAL2_BAUD	115		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 500:500000 921:921600 1500:1500000	The baud rate of the Telem2 port. The APM2 can support all baudrates up to 115, and also can support 500. The PX4 can support rates of up to 1500. If you setup a rate you cannot support or…
SERIAL2_PROTOCOL	11		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Lidar 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 12:Aerotenna uLanding 13:Beacon	Control what protocol to use on the Telem2 port. Note that the Frsky options require external converter hardware. See the wiki for details.
SERIAL3				
SERIAL3_BAUD	38		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 500:500000 921:921600 1500:1500000	The baud rate used for the Serial 3 (GPS). The APM2 can support all baudrates up to 115, and also can support 500. The PX4 can support rates of up to 1500. If you setup a rate you cannot suppor…
SERIAL3_PROTOCOL	5		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Lidar 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 12:Aerotenna uLanding 13:Beacon	Control what protocol Serial 3 (GPS) should be used for. Note that the Frsky options require external converter hardware. See the wiki for details.
SERIAL4				
SERIAL4_BAUD	115		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 500:500000 921:921600 1500:1500000	The baud rate used for Serial4. The APM2 can support all baudrates up to 115, and also can support 500. The PX4 can support rates of up to 1500. If you setup a rate you cannot support or…
SERIAL4_PROTOCOL	-1		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Lidar 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 12:Aerotenna uLanding 13:Beacon	Control what protocol Serial4 port should be used for. Note that the Frsky options require external converter hardware. See the wiki for details.
SERIAL5				
SERIAL5_BAUD	57		1:1200 2:2400 4:4800 9:9600 19:19200 38:38400 57:57600 111:111100 115:115200 500:500000 921:921600 1500:1500000	The baud rate used for Serial5. The APM2 can support all baudrates up to 115, and also can support 500. The PX4 can support rates of up to 1500. If you setup a rate you cannot support or…
SERIAL5_PROTOCOL	-1		-1:None 1:MAVLink1 2:MAVLink2 3:Frsky D 4:Frsky SPort 5:GPS 7:Alexmos Gimbal Serial 8:SToRM32 Gimbal Serial 9:Lidar 10:FrSky SPort Passthrough (OpenTX) 11:Lidar360 12:Aerotenna uLanding 13:Beacon	Control what protocol Serial5 port should be used for. Note that the Frsky options require external converter hardware. See the wiki for details.

(2) Set the obstacle avoidance sensor protocol to TeraRangerTower (PRX_TYPE is set to 3), as shown below:



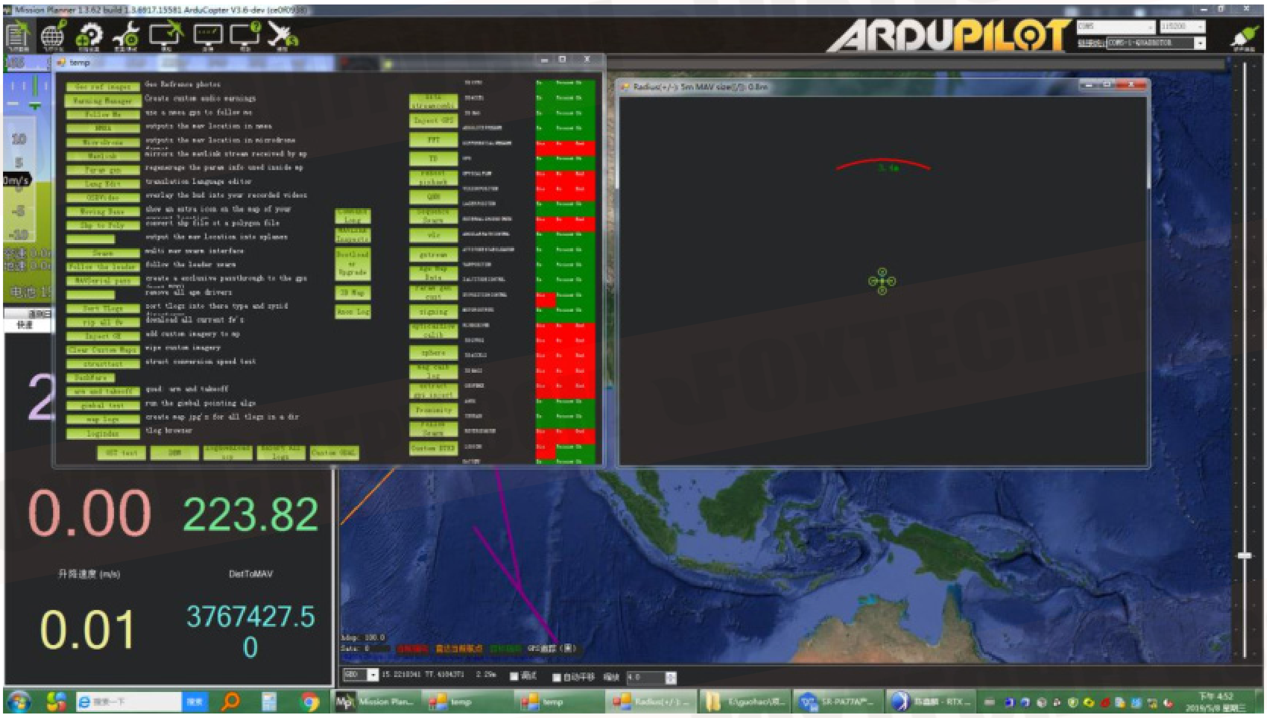
```
Command        Value  Unit     Range                                                              Description
■ PRX
PRX_IGN_ANG1   0      degrees  0 360                                                              Proximity sensor ignore angle 1
PRX_IGN_ANG2   0      degrees  0 360                                                              Proximity sensor ignore angle 2
PRX_IGN_ANG3   0      degrees  0 360                                                              Proximity sensor ignore angle 3
PRX_IGN_ANG4   0      degrees  0 360                                                              Proximity sensor ignore angle 4
PRX_IGN_ANG5   0      degrees  0 360                                                              Proximity sensor ignore angle 5
PRX_IGN_ANG6   0      degrees  0 360                                                              Proximity sensor ignore angle 6
PRX_IGN_WID1   0      degrees  0 45                                                               Proximity sensor ignore width 1
PRX_IGN_WID2   0      degrees  0 45                                                               Proximity sensor ignore width 2
PRX_IGN_WID3   0      degrees  0 45                                                               Proximity sensor ignore width 3
PRX_IGN_WID4   0      degrees  0 45                                                               Proximity sensor ignore width 4
PRX_IGN_WID5   0      degrees  0 45                                                               Proximity sensor ignore width 5
PRX_IGN_WID6   0      degrees  0 45                                                               Proximity sensor ignore width 6
PRX_ORIENT     0               0:Default 1:Upside Down                                            Proximity sensor orientation
PRX_TYPE       3               0:None 1:LightWareSF40C 2:MAVLink 3:TeraRangerTower 4:RangeFinder  What type of proximity sensor is connected
PRX_YAW_CORR   0      degrees  -180 180                                                           Proximity sensor yaw correction
```

(3) Set the obstacle avoidance type AVOID_ENABLE to UseProximitySensor, the obstacle avoidance maximum tilt angle is 10 degrees (AVOID_ANGLE_MAX is set to 1000), the obstacle avoidance action mode is stop (AVOID_BEHAVE is set to 1), and the obstacle avoidance distance in gps mode is 3m (AVOID_MARGIN is set to 3), the obstacle avoidance distance in fixed height mode is 10m (AVOID_DIST_MAX is set to 10), as shown below:



```
Command          Value  Unit      Range                                                                  Description
■ ADSB
ADSB_ENABLE      0                0:Disabled 1:Enabled                                                   Enable ADS-B
■ ANGLE
ANGLE_MAX        4500   Centi-d…  1000 8000                                                              Maximum lean angle in all flight modes
■ ARMING
■ ATC
■ AUTOTUNE
AVD_ENABLE       0                0:Disabled 1:Enabled                                                   Enable Avoidance using ADSB
■ AVOID
AVOID_ANGLE_MAX  1000             0 4500                                                                 Max lean angle used to avoid obstacles while in non-GPS modes
AVOID_BEHAVE     1
AVOID_DIST_MAX   10     meters    3 30                                                                   Distance from object at which obstacle avoidance will begin in non-GPS modes
AVOID_ENABLE     2                0:None 1:StopAtFence 2:UseProximitySensor 3:All                        Enabled/disable stopping at fence
AVOID_MARGIN     3      meters    1 10                                                                   Vehicle will attempt to stay at least this distance (in meters) from objects while in GPS
                                                                                                         modes
■ BATT
BATT2_MONITOR    0                0:Disabled 3:Analog Voltage Only 4:Analog Voltage and Current 5:Solo 6:Bebop 7:SMBus-Maxell   Controls enabling monitoring of the battery's voltage and current
■ BCN
■ BRD
BTN_ENABLE       0                0:Disabled  1:Enabled                                                  This enables the button checking module. When this is disabled the parameters for setting
                                                                                                         button inputs are not visible
■ CAM
■ CAN
CH10_OPT         0                0:Do Nothing  2:Flip  3:Simple Mode  4:RTL  5:Save Trim  7:Save WP  9:Camera Trigger  10:RangeFinder  11:Fence
                                  13:Super Simple Mode  14:Acro Trainer  15:Sprayer  16:Auto  17:AutoTune  18:Land  19:Gripper  21:Parachute Enabl···   Select which function is performed when CH10 is above 1800 pwm
CH11_OPT         0                0:Do Nothing  2:Flip  3:Simple Mode  4:RTL  5:Save Trim  7:Save WP  9:Camera Trigger  10:RangeFinder  11:Fence
                                  13:Super Simple Mode  14:Acro Trainer  15:Sprayer  16:Auto  17:AutoTune  18:Land  19:Gripper  21:Parachute Enabl···   Select which function is performed when CH11 is above 1800 pwm
CH12_OPT         0                0:Do Nothing  2:Flip  3:Simple Mode  4:RTL  5:Save Trim  7:Save WP  9:Camera Trigger  10:RangeFinder  11:Fence
                                  13:Super Simple Mode  14:Acro Trainer  15:Sprayer  16:Auto  17:AutoTune  18:Land  19:Gripper  21:Parachute Enabl···   Select which function is performed when CH12 is above 1800 pwm
CH7_OPT          0                0:Do Nothing  2:Flip  3:Simple Mode  4:RTL  5:Save Trim  7:Save WP  9:Camera Trigger  10:RangeFinder  11:Fence
                                  13:Super Simple Mode  14:Acro Trainer  15:Sprayer  16:Auto  17:AutoTune  18:Land  19:Gripper  21:Parachute Enabl···   Select which function is performed when CH7 is above 1800 pwm
CH8_OPT          40               0:Do Nothing  2:Flip  3:Simple Mode  4:RTL  5:Save Trim  7:Save WP  9:Camera Trigger  10:RangeFinder  11:Fence
                                  13:Super Simple Mode  14:Acro Trainer  15:Sprayer  16:Auto  17:AutoTune  18:Land  19:Gripper  21:Parachute Enabl···   Select which function is performed when CH8 is above 1800 pwm
CH9_OPT          0                0:Do Nothing  2:Flip  3:Simple Mode  4:RTL  5:Save Trim  7:Save WP  9:Camera Trigger  10:RangeFinder  11:Fence
                                  13:Super Simple Mode  14:Acro Trainer  15:Sprayer  16:Auto  17:AutoTune  18:Land  19:Gripper  21:Parachute Enabl···   Select which function is performed when CH9 is above 1800 pwm
CHUTE_ENABLED    0                0:Disabled 1:Enabled                                                   Parachute release enabled or disabled
■ CIRCLE
■ COMPASS
DEV_OPTIONS      0                                                                                       Bitmask of developer options. The meanings of the bit fields in this parameter may vary at
                                                                                                         any time. Developers should check the source code for current meaning
DISARM_DELAY     10     Seconds   0 127                                                                  Delay before automatic disarm in seconds. A value of zero disables auto disarm.
■ EK2
EK3_ENABLE       0                0:Disabled  1:Enabled                                                  This enables EKF3. Enabling EKF3 only makes the maths run, it does not mean it will be used
                                                                                                         for flight control. To use it for flight control set AHRS_EKF_TYPE=3. A reboot or restart···
                                  0:Normal Start-up  1:Start-up in ESC Calibration mode if throttle high  2:Start-up in ESC Calibration mode   Controls whether ArduCopter will enter ESC calibration on the next restart.  Do not adjust
```

(4) After setting the above parameters and saving, then restart flight control, connect MissionPlanner, CTRL+F to open the debugging window to see the obstacle avoidance distance, as shown below: (target is in 0 sector 3.4m)



After the above settings are completed, then can perform the outdoor test. When the drone is less than 3m away from the obstacle in gps mode (less than 10m in the fixed height mode), the drone will have a brake action, and the joystick cannot make the drone continue to fly forward.

FAQ

Q: Can 77G radar only avoid obstacles in one direction?

A: In the open source flight control platform, this 77G radar currently only supports obstacle avoidance in one direction (forward obstacle avoidance); in the DJI flight control platform, this 77G radar can output data of multiple sectors, and subsequently cooperate with Daxiang Zhiyin, that support obstacle avoidance in multi-direction, and it coming soon.

This content is subject to change.

Download the latest version from

https://www.foxtechfpv.com/77ghz-millimeter-wave-obstacle-avoidance-radar.html

For everyday updates, please follow

Foxtech Facebook page: https://www.facebook.com/foxtechhobby

YouTube Channel: https://www.youtube.com/user/foxtechonline/featured?view_as=subscriber