

Viewpro Gimbal Camera Protocol V3.1

Date: 2021.3.03

Update Records:

2021.03.03	Add angle automatic feedback command, add query angle/zoom time usage notes

Content

1. Serial Control Connection:	3
2. CMD_CONTROL -control gimbal Movement:	4
2.1) CMD_GET_ANGLES_EXT: 3e 3D 00 3D 00.....	8
2.2) Other commands:.....	9
3. Camera_control_cmd:	11
3.1) Z10F, Z18F, Z30F, Q10F, Q18F, Q30F command:.....	11
3.2) Q30XIR command:.....	11
3.3) QIR19 command:.....	12
3.4) Z6KA7 /Z5S/Z-A7 command.....	12
3.5) Z25K/Z40K command.....	13
3.6) Z10N/Z30N command.....	13
3.7) T serials model:.....	13
3.7.1) Record/photograph command:(TIR/TL/TM/TIRM/IR..T).....	13
3.7.2) Normal tracking: (TIR/TL/TM/TIRM/IR..T).....	13
3.7.2.1) Target_tracking.....	14
3.7.2.2) Stop_tracking.....	14
3.7.2.3) Change target,move cursor to the new target.....	14
3.7.2.4) Finger tracking mode.....	14
3.7.3) SD card inquiry command and feedback command protocol: (TIR/TL/TM/TIRM/IR..T).....	15
3.7.4) Picture in picture: (TIR).....	16
3.7.5) Color change:(TIR/IR..T).....	16
3.7.6) Thermal camera Digital Zoom cmd: (TIR/IR..T).....	17
3.7.7) laser light cmd: (TL).....	17
3.7.8) Laser Ranger Finder command: (TM/TIRM).....	17
4. Camera settings command: (blue byte for different setting value)	18
5. How to connect gimbal to Pixhawk with serial port ?	21
6. Viewpro Gimbal Supported Mavlink msg ID List	23

1. Serial Control Connection:

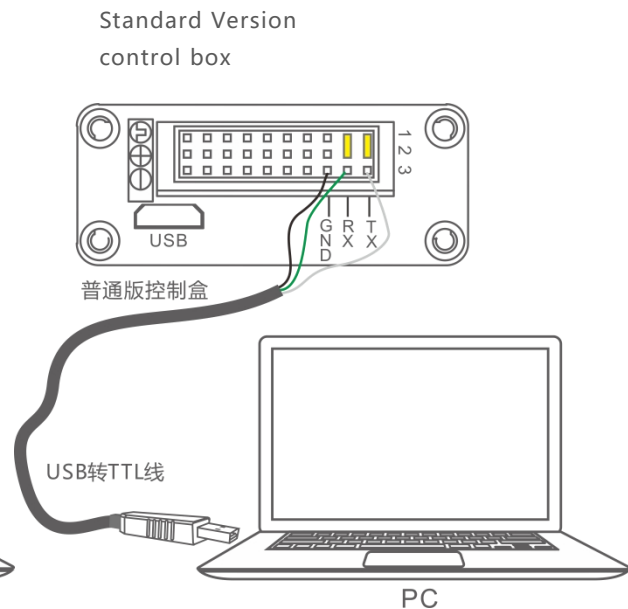
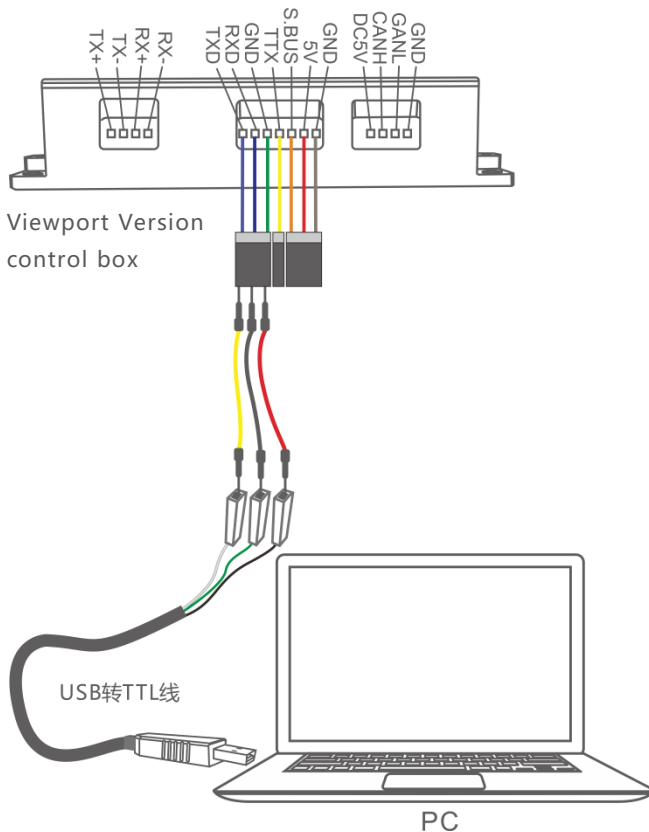
TTL 3.3v UART baud: 115200, 8/1/N, HEX

Use 'USB to TTL' cable to connect

White cable (the 'RX' of cable) ----- TX

Green cable (the 'TX' of cable) ----- RX

Black cable (the 'GND' of cable) ----- GND



Please connect your RX to TXD, and TX to RXD as shown in the figures.

Note:

After sending "AA 55 08 06 FF" , TX2(TTX), RX2(S.BUS), GND will be the second serial port of the gimbal, UART baud: 115200

RX1/ RX2 and TX1 / TX2 must be connected with yellow jumper caps before connecting UART/FTDI devices. Please connect your RX to TX, and TX to RX as shown in the figures.

CAUTION!!

- 1) the signals inside red frame are all TTL signals, DO NOT connect these pin headers to any power(VCC) or ground(GND)!
- 2) The other signals is for PWM input signals to control the gimbal,
 PWM in: connect to your PWM receiver signals (does not include AV, AV output is for CVBS video signal)
 5V out: for your PWM receiver power supply.
 GND: connect to your PWM receiver GND.

2. CMD_CONTROL -control gimbal Movement:

FF 01 0F 10 RM PM YM Rsl Rsh Ral Rah Psl Psh Pal Pah Ysl Ysh Yal Yah CS
 HEAD CTRL_MODE SPEED ANGLE SPEED ANGLE SPEED ANGLE CHECKSUM

RM: Roll control Mode, **00**=MODE_NO_CTRL, **01**=MODE_SPEED, **02** = MODE_ANGLE, **03** = MODE_SPEED_ANGLE, **04** = MODE_RC, **05**=MODE_ANGLE_REF_FRAME, **06** = MODE_RC_HIGH_RES.

ROLL SPEED: RS= **0xRshRsl** Rsl = low byte; Rsh= high byte (2 byte signed, little-endian order), units: 0,1220740379 degree/sec;

ROLL ANGLE: RA= **0xRahRal** Ral = low byte ; Rah= high byte (2 byte signed, little-endian order), units: 0.02197265625 degree/sec;

PM Psl Psh Pal Pah: Pitch control, same as roll control formats

YM Ysl Ysh Yal Yah: Yaw control, same as roll control formats

sl= Speed Low byte, sh = speed high byte, al= angle low byte, ah= angle high byte.

If use RC mode, PA/YA is RC value should be -500~500, (-500 is PWM 1000ms, 0 is PWM 1500ms, 500 is PWM 2000ms)

CS = body checksum, checksum is calculated as a sum of bytes from 'RM' to 'YAH' modulo 256.

example1: ROLL no control, PITCH speed mode 1.22degree/sec, YAW speed mode 1.22degree/sec.

FF 01 0F 10 00 01 01 00 00 00 00 0A 00 00 00 0A 00 00 00 16

Example2: ROLL no control, PITCH angle mode to 40 degree down REF home position, YAW angle mode 40 degree left REF home position.

FF 01 0F 10 00 05 05 00 00 00 00 00 00 1C 07 00 00 E4 F8 09

Example3: RC control pitch down(PWM = 1920, PA value = 1920-1500 = 420), RC control Yaw left (PWM = 1050, YA value = 1050-1500 = -450)

FF 01 0F 10 00 04 04 00 00 00 00 00 00 A4 01 00 00 3E FE E9

Example4: at all mode, stop command pitch and yaw use the command:

FF 01 0F 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Name	Type	Min	Max	Possible values, remarks
<i>Legacy format: mode is common for all axes</i>				
CONTROL_MODE	1u			<p>Bits 0..3 for mode, bits 4..7 for flags.</p> <p>Modes:</p> <p>MODE_NO_CONTROL=0</p> <p>If this mode is set for all axes, finish serial control and restore normal RC control. If set for single axis, does not change its current control mode.</p> <p>MODE_SPEED=1</p> <p>Camera travels with the given speed in the Euler coordinates until the next CMD_CONTROL command comes. Given angle is ignored.</p> <p>MODE_ANGLE=2*</p> <p>Camera travels to the given Euler angle with the fixed speed. Speed is decreased near target to keep control smooth. Low-pass filter may be applied for the same reason.</p> <p>MODE_SPEED_ANGLE=3</p> <p>Camera travels with the given speed. Additionally, controller keeps the given angle and fix accumulated error by the outer PI-loop. This mode allows the most precise type of control (see fig.1 for example), but it requires pretty fast update rate to keep it smooth, or apply low-pass filtering for speed and angle.</p> <p>MODE_RC=4*</p> <p>The ANGLE parameter is used as RC signal and overrides any other signal source, assigned to this axis. Normal working range is -500..500. A special value -10000 encodes a "signal lost" condition.</p> <p>The flag CONTROL_FLAG_AUTO_TASK can affect this mode (see below).</p> <p>MODE_RC_HIGH_RES=6*</p> <p>The same as the MODE_RC, but the range of the ANGLE parameter has better resolution:</p> <p>-16384..16384. A special value -32768 encodes a "signal lost" condition.</p>

				<p>MODE_ANGLE_REL_FRAME=5*</p> <p>First, the neutral point of a camera relative to a frame is found in the Euler coordinates for a given axis. Then, the given angle (in $\pm 360^\circ$ range) is added to this point, and camera travels to it. Note that the given angle does not relate to a particular motor, it relates to global Euler angles!</p> <p>Flags:</p> <p>CONTROL_FLAG_AUTO_TASK=(1<<6)</p>
--	--	--	--	---

				<p>- If mode is one of the <MODE_ANGLE, MODE_ANGLE_REL_FRAME>, the task is processed with the speed and acceleration configured for <i>automated tasks</i>. If the SPEED parameter is provided, it's used instead. When all target angles are reached with the 1- degree tolerance, confirmation is sent: CMD_CONFIRM(CMD_CONTROL, 1). Use this flag to move gimbal to a certain position as fast as possible, and receive confirmation when the target is reached.</p> <p>CONTROL_FLAG_FORCE_RC_SPEED=(1<<6)</p> <p>- If mode is MODE_RC, this flag forces a control in the "SPEED" mode, with the dead-band, trimming and inversion settings are NOT applied to the provided RC signal, but the LPF, Expo curve and ACC limiter are still applied. Use this flag to control gimbal from remote applications, where signal is well-defined and you need to have a direction of rotation that does not depend on gimbal's "Inverse" and "Mode" parameters.</p> <p>CONTROL_FLAG_HIGH_RES_SPEED=(1<<7)</p> <p>Speed units changed to 0.001 deg/sec for extremely slow motion (like timelapse shooting)</p> <p><i>* In the control modes "MODE_ANGLE", "MODE_RC", and "MODE_ANGLE_REL_FRAME", if the "SPEED" parameter is not provided (set to zero), the speed is defined by the RC settings. Relationship: SPEED = settings.RC_SPEED*16.</i></p>
<i>Extended format : mode is set independently for each axes</i>				
	CONTROL_MODE[3]	1u*3		see definition above
<i>The remaining part is common for all formats</i>				
axis = (1..3)	SPEED	2s	- - -	- - -
				<p>Speed of rotation. Overrides the speed settings in the GUI and from the adjustable variables.</p> <p>Notes:</p> <ul style="list-style-type: none"> If the acceleration limiter is enabled in the RC settings, the actual speed is filtered by it;

CMD_GET_ANGLES_EXT - Information about angles in different format

Name	Type	Min	Max	Possible values, remarks
axis = (1..3)	IMU_ANGLE	2s		IMU angles in 14-bit resolution per full turn <i>Units: 0,02197265625 degree</i>
	TARGET_ANGLE	2s		Target angles, in 14-bit resolution <i>Units: 0,02197265625 degree</i>
	STATOR_ROTOR_ANGLE	4s		Relative angle of joints between two arms of gimbal structure, measured by encoder or 2 nd IMU. Value 0 corresponds to normal position of a gimbal. This angle does not overflow after multiple turns. <i>Units: 0,02197265625 degree</i>
	RESERVED	10b		

2.1.2)Angles Automatic Feedback Setting

Open gimbal' s attitude continuous feedback: 3E 3E 00 3E 00

Close gimbal' s attitude continuous feedback: 3E 3D 00 3D 00

Command:

3E 55 15 6A 58 E8 03 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 50

Description:

3E 55 15 6A **Frame header**

E8 03 = 0X03E8 interval time(1000ms interval, the time interval for the gimbal auto return to the angle value)

50 last byte is low byte of sum of all red bytes.

The example of the gimbal posture data return to data:

3E 58 14 6C F0 84 00 00 8C 03 FB FF 00 00 00 00 00 00 00 00 E9 FF 69 03 38 04 8D

3E 58 14 6C return data frame header

F0 84 is the time stamp, ms is the unit

8d is the checksum of the data in red

The data structure as below:

```
typedef struct {
    uint16_t timestamp; -----0X84F0 unit: ms
    int16_t rollIMUangle;-----0X0000 unit: 0.02197 degree
    int16_t pitchIMUangle;-----0X038C unit: 0.02197 degree
    int16_t yawIMUangle; -----0XFFFB unit: 0.02197 degree
    int16_t rollIMUspeed;-----0X0000 unit: 0.06103 degree/sec
```

```

int16_t pitchIMUSpeed;-----0X0000 unit: 0.06103 degree/sec
int16_t yawIMUSpeed; -----0X0000 unit: 0.06103 degree/sec
int16_t rollStatorRotorAngle; -Absolute angle unit: 0.02197 degree
int16_t pitchStatorRotorAngle; Absolute angle unit: 0.02197 degree
int16_t yawStatorRotorAngle; Absolute angle unit: 0.02197 degree
} T_GimbalGetAnglesExtReq;
Angle unit: 0.02197

```

2.2) Other commands:

Return head:3E 45 01 46 12 12	Look down: 3E 45 01 46 11 11
Center yaw:3E 45 01 46 23 23	Restore Settings: 3E 45 01 46 35 35
Motor on: 3e 45 01 46 0b 0b	Motor off: 3e 45 01 46 0c 0c
follow_yaw_disable: 3E 1F 06 25 01 1F 00 00 00 00 20	follow_yaw_enable: 3E 1F 06 25 01 1F 01 00 00 00 21
query_follow_state: 3E 40 02 42 01 1F 20	save gimbal settings: 3E 20 00 20 00 00

3. Camera_control_cmd: different model use different commands, so please use command according to your actual model

3.1) Z10F, Z18F, Z30F, Q10F, Q18F, Q30F command:

zoom out: 81 01 04 07 37 FF	zoom in: 81 01 04 07 27 FF	stop zoom: 81 01 04 07 00 FF
focus in: 81 01 04 08 27 FF	focus out: 81 01 04 08 37 FF	stop focus: 81 01 04 08 00 FF
Zoom_direct_pos: 81 01 04 47 0p 0q 0r 0s FF		
Query_zoom_position: 81 09 04 47 ff (Query frequency can not be too high, interval should be at least 50ms)		
Zoom_position_feedback: 90 50 0p 0q 0r 0s FF 0xpqrs: zoom_position_value		
photograph /record action command: 81 01 04 68 xx FF xx = : 01 photograph 02 start record 03 stop record 04 invert record state 05 switch to record mode or picture mode Inquiry record/photograph state: 81 09 04 68 FF Camera Feedback: 81 09 04 68 00 FF stop record 81 09 04 68 01 FF recording 81 09 04 68 10 FF photograph mode		
day mode: 81 01 04 01 03 ff		
night mode: 81 01 04 01 02 ff		
day/night auto switch mode: 81 01 04 51 02 ff		

3.2) Q30XIR command:

Switch Modes: f8 00 12 01 01 00 00 14 VL(Visible Light) only f8 00 12 01 02 00 00 15 IR only f8 00 12 01 03 00 00 16 Fusion-A f8 00 12 01 04 00 00 17 Fusion-B f8 00 12 01 05 00 00 18 PIP f8 00 12 01 06 00 00 19 PIP-A f8 00 12 01 07 00 00 1a PIP-B
IR Color Palette Settings: f8 00 12 02 00 00 00 14 Iron red f8 00 12 02 01 00 00 15 Amber f8 00 12 02 02 00 00 16 Yellow f8 00 12 02 03 00 00 17 Green blue yellow red f8 00 12 02 04 00 00 18 Black red f8 00 12 02 05 00 00 19 Blue yellow red f8 00 12 02 06 00 00 1a Blue red yellow f8 00 12 02 07 00 00 1b Blue green f8 00 12 02 08 00 00 1c White hot f8 00 12 02 09 00 00 1d Blue green red f8 00 12 02 0a 00 00 1e Red warning f8 00 12 02 0b 00 00 1f Amer-2

Record Operations:
f8 00 82 01 00 00 00 83 Start record (AVI)
f8 00 84 00 00 00 00 84 Stop record
f8 00 80 00 00 00 00 80 Snapshot (RAW)
f8 00 80 01 00 00 00 81 Snapshot (JPEG)
Adjust Display:
f8 00 12 03 07 00 00 1c
Zoom in circle
f8 00 12 03 08 00 00 1d Zoom 1x
f8 00 12 03 08 01 00 1e Zoom 2x
f8 00 12 03 08 02 00 1f Zoom 3x
f8 00 12 03 08 03 00 20 Zoom 4x

3.3) QIR19 command:

IR Color Palette Settings:
f8 00 22 03 00 00 00 25 Iron red
f8 00 22 03 01 00 00 26 Amber
f8 00 22 03 02 00 00 27 Yellow
f8 00 22 03 03 00 00 28 Green blue yellow red
f8 00 22 03 04 00 00 29 Black red
f8 00 22 03 05 00 00 2a Blue yellow red
f8 00 22 03 06 00 00 2b Blue red yellow
f8 00 22 03 07 00 00 2c Blue green
f8 00 22 03 08 00 00 2d White hot
f8 00 22 03 09 00 00 2e Blue green red
f8 00 22 03 0a 00 00 2f Red warning
f8 00 22 03 0b 00 00 30 Amer-2
Record Operations:
f8 00 82 01 00 00 00 83 Start record (AVI)
f8 00 84 00 00 00 00 84 Stop record
f8 00 80 00 00 00 00 80 Snapshot (RAW)
f8 00 80 01 00 00 00 81 Snapshot (JPEG)
Zoom in circle
f8 00 30 01 00 00 00 31 Zoom 1x
f8 00 30 02 00 00 00 32 Zoom 2x
f8 00 30 03 00 00 00 33 Zoom 3x
f8 00 30 04 00 00 00 34 Zoom 4x

3.4) Z6KA7 /Z5S/Z-A7 command

Power on/off:FF 01 00 50 00 00 51	zoom_wide:FF 01 00 40 00 00 41	zoom_tele: FF 01 00 20 00 00 21
stop_zoom: FF 01 00 00 00 00 01	Picture: FF 01 00 07 00 66 6E	Record: FF 01 00 07 00 55 5D
Focus: FF 01 00 80 00 00 81	focus hold: FF 01 01 80 00 00 82	focus release:FF 01 02 80 00 00 83
shoot_under_focus_hold : FF 01 00 07 00 67 6F	picture_interval_2s_time: FF 01 00 07 02 6f 79	picture_interval_3s_time: FF 01 00 07 03 6f 7A
picture_interval_OFF: FF 01 00 07 00 6f 77		

3.7.2.1) Target_tracking

Target_tracking:

```
7e 7e 44 00 00 71 fe 00 00 00 00 01 01 3c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 ed
```

3.7.2.2) Stop_tracking

Stop_tracking:

```
7e 7e 44 00 00 26 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 66
```

3.7.2.3) Change target,move cursor to the new target

Step1: First send Re-select_target[] ; entering the re-select target mode

Re-select_target[]=

```
7e 7e 44 00 00 71 fe 00 00 00 00 00 01 3c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 ec
```

Step 2: move the cursor x, y. speed.

Protocol: cmd_data[5]=0X71 // tracking use '+' cursor cmd_data[11] = 0X00

cmd_data[7]= 0xXL, X LOW BYTE

cmd_data[8]= 0xXH, X HIGT BYTE

cmd_data[9]= 0xYL, Y LOW BYTE

cmd_data[10]= 0xYH, Y HIGH BYTE

Down_cursor_2[]=

```
7e 7e 44 00 00 71 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 b3
```

3.7.2.4) Finger tracking mode

cmd_data[5]=0X86, cmd_data[11] = 0X01, cmd_date[13] = track size

Cmd_data[7]=0xXL, XLOW BYTE

Cmd_data[8]=0xXH, XHIGT BYTE

Cmd_data[9]=0xYL, YLOW BYTE

Cmd_data[10]=0xYH, YHIGH BYTE

Track the target at position: (769, 769)

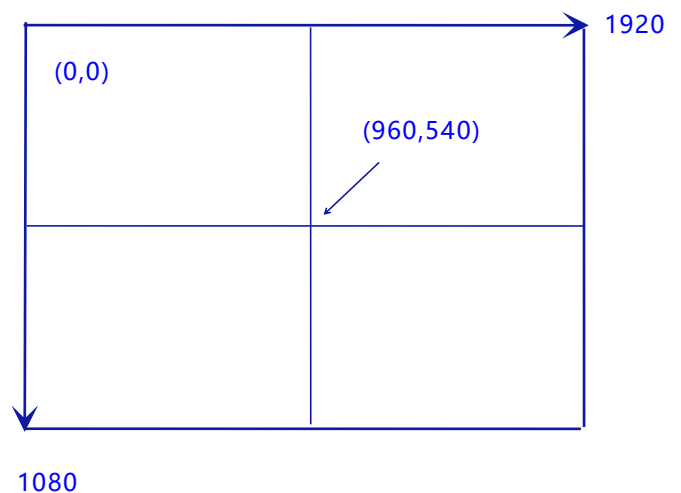
```
7e 7e 44 00 00 86 00 01 03 01 03 01 00 3c 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 0b
```

Track the target at position: (1025, 769)

```
7e 7e 44 00 00 86 00 01 04 01 03 01 00 3c 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 0c
```

Track the target at position: (257, 257)

```
7e 7e 44 00 00 86 00 01 01 01 01 01 00 3c 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 07
```



3.7.3) SD card inquiry command and feedback command protocol: (TIR/TL/TM/TIRM/IR..T)

7E	7E	44	00	00	Data5	Data6	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	CS

- Inquiry command to get information. 48bytes
Header: 7e 7e 44 CS= sum of all other bytes, mod 256.

- ❖ format SD card Data5 = 0x8a Data6 = 00
- ❖ Inquiry SD card information:
 - Data5 = 0x8b
 - Data6 =
 - 02-inquiry SD card status
 - 03-inquiry SD card total capacity
 - 04-inquiry SD card remain capacity
 - 05-inquiry SD card remain pictures quantity capacity
 - 06-inquiry SD card remain video time capacity

- Feedback command. 8bytes

Data0=BB	Data1=09	Data2	Data3	Data4	Data5	Data6	Data7=CS
----------	----------	-------	-------	-------	-------	-------	----------

- Data2 = 1 : SD card status,
- Data2 = 2 : SD card total capacity
- Data2 = 3 : SD card remain capacity
- Data2 = 4 : SD card remain pictures quantity
- Data2 = 5 : SD card remain video time
- ◆ Feedback command details ([0] means bit0)

1) SD card status

BB-----header1 09-----header2 Data2 = 01-----SD card status,
Data 3:
[0] : Is inserted
[1] : Is initializing
[2] : Is read only
[3] : Is formatted
[4] : Is formatting
[5] : Is full
[6] : Is verified
[7] : Is invalid format
Data4:
[0] : Has error
[1] : 1: Is Recording 0: not recording
[2-7] Reserved
Data5 res
Data6 res
Data7 cs

2) SD card total capacity

BB-----header1 09-----header2 Data2 = 02-----SD card total capacity
Data3= ss, Data4=rr, Data5=qq, Data6=pp uint32
0xppqqrrss = SD car capacity(unit MB,LSB is at the front and MSB is at the back)
Data7 cs

3.7.6) Thermal camera Digital Zoom cmd: (TIR/IR..T)

Protocol: cmd_data[5]=0X7d

Cmd_data[6]= 0x81: 1x, 0x82: 2x, 0x83: 3x, 0x84: 4x

IR_DZOOM_1X:

7e 7e 44 00 00 **7D 81** 00
00 00

IR_DZOOM_2X:

7e 7e 44 00 00 **7D 82** 00
00 00

3.7.7) laser light cmd: (TL)

3.5.6.1) Z30TL

Zoom_in_laser: FF 01 01 04 00 1a 20	Zoom_out_laser: FF 01 01 04 01 1a 21
Laser_on: FF 01 01 01 01 00 04	Laser_off: FF 01 01 01 00 00 03

3.5.6.2) Z10TL

zoom_in_laser: FF 01 00 20 00 00 21	zoom_out_laser: FF 01 00 40 00 00 41	zoom_stop_laser: FF 01 00 00 00 00 01
laser_on: FF 01 00 09 00 01 0b	laser_off: FF 01 00 0b 00 01 0d	

3.7.8) Laser Ranger Finder command: (TM/TIRM)

(Z10TIRM, WK10TIRM, Q10TIRM, Q30TIRM, Z30TM)

Read the ranging result :

Host send:		
10 83 00 7d	one time get LRF data	
10 83 40 3d	continuous get LRF data	
10 84 7C	stop get LRF data	
LRF feedback data:		
Identifier	parameter	Description
0x01	Byte1	D7: Invalid distance data flag, 0 = valid data, 1 = Invalid data D6: Invalid inclination data flag, 0 = valid data, 1 = Invalid data D5: Distance data accuracy, 0 = 0.5/LSB, 1 = 0.1/LSB D4: Data unit, 0 = Meter, 1 = code D3 - D0: Operating mode, 0 = Ranging, 1 = Speed measurement
	Byte2	When the data is valid: Distance[D15..D8] When the data is invalid: 0x80 = invalid distance data. 0x81=system error
	Byte3	When the data is valid: Distance[D7..D0] When the data is invalid: reserved.
	Byte4	inclination: Angle[D7..D0], Binary complement format. Do not send inclination angle during speed measurement.
Example:		
1、LRF feedback data: 0x10 <u>0x01</u> <u>0x20</u> <u>0x12</u> <u>0xD7</u> <u>0xFE</u> 0xF8 distance = 0x12D7 = 4823 = 482.3m pitch angle = 0xFE = -2°		

2. LRF feedback data: 0x10 0x01 0xA0 0x80 0x00 0x01 0xDE
invalid distance data(byte4 = 0xA0, bit7 = 1) pitch angle = 1°

Last byte is Checksum = 0x00- byte2-byte3-byte4-byte5

4. Camera settings command: (blue byte for different setting value)

- 4.1) AA 55 04 **08** FF timezone set, beijing is 8
- 4.2) AA 55 05 **03** FF OSD on/Off (refer [table A](#))
- 4.3) AA 55 06 **0F** FF Magnetic Variation set <https://skyvector.com/> <http://www.magnetic-declination.com/>
- 4.4) AA 55 07 **11** FF OSD input enable/disable (refer [table B](#))
- 4.5) AA 55 08 **02** FF baudrate setting (refer [table C](#))
- 4.6) AA 55 09 **01** FF EO digital zoom : 1 on 0: off
- 4.7) AA 55 0A **96** FF temperature alarm low byte
- 4.8) AA 55 0B **00** FF temperature alarm high byte
- 4.9) AA 55 0C **01** FF 1: track enable 0: track disable
- 4.10) AA 55 0D **01** FF 0: stop LRF 1: 1HZ get LRF data 2: continuously get LRF data 3: one time get LRF data
- 4.11) AA 55 0E **01** FF 1: 4k25fps 2: 1080p25 Z40K/Q40TIR record definition set
- 4.12) AA 55 0F **01** FF 0: OSD GPS is UAV 1: OSD GPS is target
- 4.13) AA 55 10 **00** FF s.bus/mavlink channels map set 1: 1~7, 2: 6~12 3: 8~14 4: custom channels, see 3.14 as following
- 4.14) AA 55 11 **YW PT MO ZM FC RP MU** FF custom channels map set. (all byte is hex data, set to 0 if no control)

AA 55 11 **YW PT MO ZM FC RP MU** **FF**
 Frame head body frame tail

Command format: (do not change red bytes, just change blue bytes as your channel setting), one byte define two channels,

bit 0-3 is chan#A 1100us < > 1500us(initial value)

bit 4-7 is chan#B 1900us < > 1500us(initial value)

AA 55 11 **YW PT MO ZM FC RP MU** **FF**

								__Multi :chan# = 0x0 M track (1900), chan# = 0x0 U stop track (1100)
								__PIC/REC: chan# = 0x0 R record (1900), chan# = 0x0 P picture (1100)
								__Focus:chan# = 0x0 F focus in (1900),chan# = 0x0 C focus out (1100)
								____ Zoom: chan# = 0x0 Z zoom in (1900),chan# = 0x0 M zoom out (1100)
								____ Mode:chan# = 0x0 M recenter (1900),chan# = 0x0 O slow speed (1100)
								____ Pitch: chan# = 0x0 P pitch down (1900),chan# = 0x0 T pitch up (1100)
								____Yaw: 0x0 Y = yaw right (1900),0x0 W = yaw left (1100)

For example: .1(yaw), 2(pitch), 3(mode), 4(zoom), 5(focus),6(picture),7(record),8(stop track),9(start track)

AA 55 11 **11 22 33 44 55 76 98** **FF**

								__ Multi :channel 8: from 1500 to 1100 channel 9: from 1500 to 1900
								__PIC/REC:channel 6: from 1500 to 1100 take a picture, channel 7: from 1500 to 1900 start/stop record
								__ Focus: channel 5: 1100 focus out, 1500 focus stop,1900 focus in
								____ Zoom:channel 4: 1100 zoom out, 1500 zoom stop, 1900 zoom in
								____ Mode:channel 3: 1100 low speed, 1500 mid speed, 1900 recenter
								____ Pitch:channel 2: 1100 pitch up, 1500 pitch stop, 1900 pitch down
								____Yaw: channel 1: 1100 yaw left, 1500 yaw stop, 1900 yaw right

channel control invert

AA 55 1A IV FF CMD[0X1a] = YPMZ_INVERT_FLAG; default IV = ff

IV bit	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
channel	YAW Right	YAW Left	PIT Down	PIT Up	Recenter	Speed adjust	Zoom in	Zoom out
Default	1: 1900	1:1100	1: 1900	1:1100	1: 1900	1: 1100 slow	1: 1900	1:1100
Invert	0: 1100	0:1900	0: 1100	0:1900	0: 1100	0: 1900 slow	0: 1100	0:1900

AA 55 1B IV FF CMD[0X1B] = FPRM_INVERT_FLAG; default IV = ff

IV bit	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
channel	Focus in	Focus out	Record start/stop	photograph	Multi high (start/stop track)	Multi low (digital zoom/manual focus)	reserved	reserved
Default	1: 1900	1:1100	1: 1900	1:1100	1: 1900	1: 1100 slow		
invert	0: 1100	0:1900	0: 1100	0:1900	0: 1100	0: 1900 slow		

- 4.15) AA 55 20 01 FF restore IP address to 192.168.2.119 (T serial model)
- 4.16) AA 55 21 10 FF Z40k/Q40TIR definition OSD on/off 10: on 20: off
- 4.17) AA 55 22 80 FF Z40k/Q40TIR date OSD on/off 80: on 90: off
- 4.18) AA 55 23 40 FF Z40k/Q40TIR OSD on/off 30: off 40: on
- 4.19) AA 55 24 A0 FF Z40k/Q40TIR SD card capacity OSD on/off A0: off B0: on
- 4.20) AA 55 25 C0 FF Z40k/Q40TIR record time OSD on/off C0: off D0 : on
- 4.21) AA 55 1E SZ FF

use track size cmd to set the size you want . just one time, the size will be storage into eeprom: AA 55 1E SZ FF, note SZ is hex

0x24: Small template 32

0x28: Medium template 64

0x30: Big template 128

The above information can be superimposed, such as:

0x2c: Small template+ Medium template

0x38: Medium template+ Big templatefor

example: you can send AA 55 1E 28 FF , for set track size to medium template

TABLE A : OSD ON/OFF settings

BIT0	0: all OSD on 1: all OSD off
BIT1	0: cross OSD on 1: cross OSD off
BIT2	0: pitch& yaw ON 1: pitch & yaw OFF
BIT3	0: XY shift osd on 1: XY shift osd off
BIT4	0: GPS OSD ON 1: GPS OSD OFF
BIT5	0: time OSD ON 1: time OSD off
BIT6	0: VL-MAG OSD ON 1: VL-MAG OSD OFF
BIT7	0: SMALL OSD WORD 1: BIG OSD WORD

Table C:

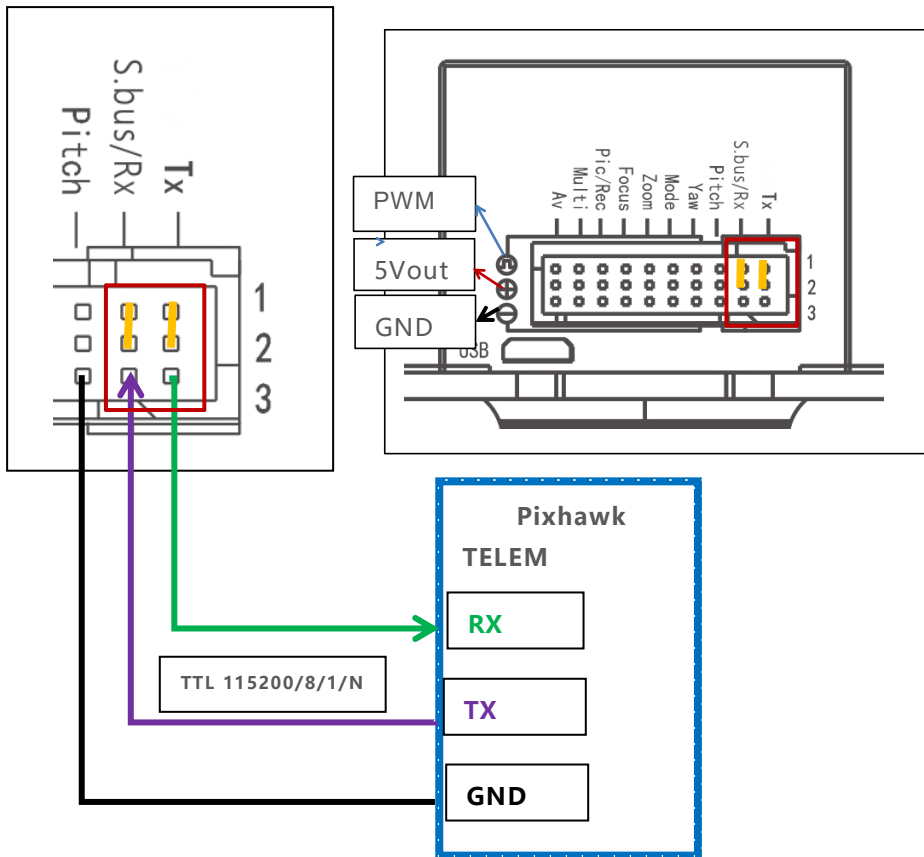
Baudrate setting	
0:	2400
1:	4800
2:	9600
3:	19200
4:	38400
5:	57600
6:	115200 (default)
7:	S.BUS mode

TABLE B: OSD input enable/disable

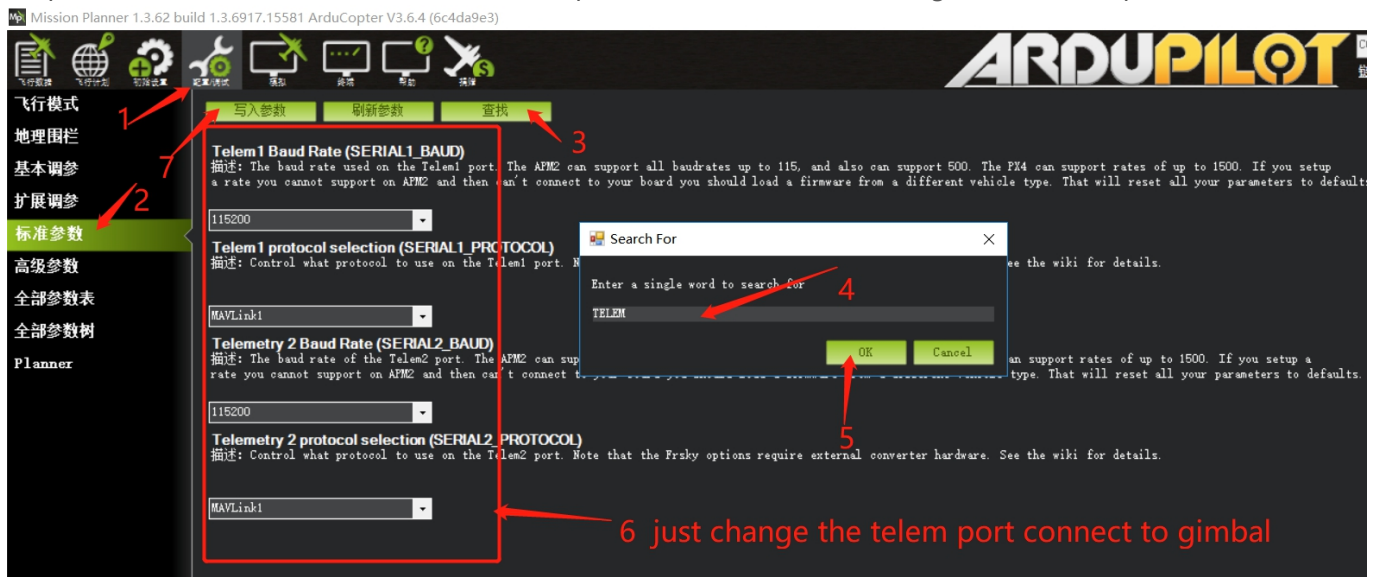
BIT0	0: not save setting 1: save
BIT1	0: time input disable 1: time input enable
BIT2	0: GPS time input disable 1: GPS time input enable
BIT3	0: GPS 1: MGRS
BIT4	0: pitch/yaw input disable 1: pitch/yaw input enable
BIT5	0: VL-MAG input disable 1: VL-MAG input enable
BIT6	0: zoom times display 1: fov display
BIT7	0: word have no black frame 1: word have black frame

5. How to connect gimbal to Pixhawk with serial port ?

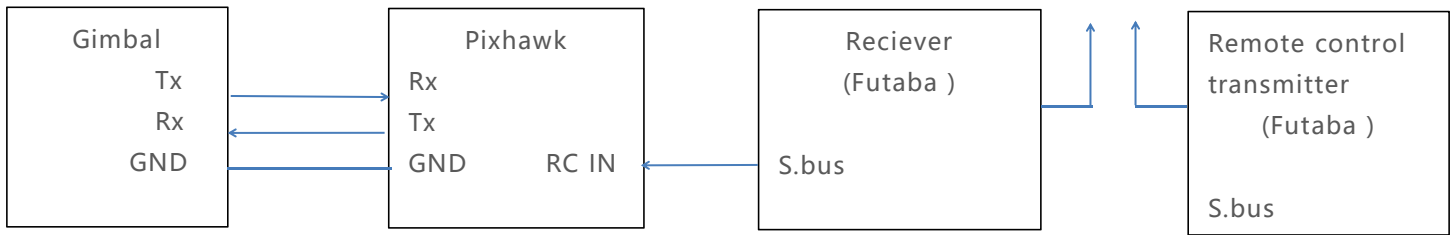
Step 1: serial port of gimbal connect to Pixhawk serial port (telem1 or telem2)



Step2: Set Pixhawk baudrate to 115200 and protocol select mavlink1. (eg, use mission planner)



Use "RCIN" connect to S.bus port of futaba reciever, user can control gimbal by futaba remote controller.



SET channels for control gimbal camera :

AA 55 10 01 FF 1~7ch

AA 55 10 02 FF 6~12ch

AA 55 10 03 FF 8~14ch

AA 55 10 04 FF custom channel mapping as following

For example: use 7~14chan to control

AA 55 11 77 88 99 AA BB CC DD FF

6. Viewpro Gimbal Supported Mavlink msg ID List

A)MAVLINK_MSG_ID_SYSTEM_TIME get time for OSD

B)MAVLINK_MSG_ID_COMMAND_LONG

1) MAV_CMD_DO_MOUNT_CONFIGURE

l) MAV_MOUNT_MODE_RETRACT ----off motor function

li) MAV_MOUNT_MODE_NEUTRAL-----home position

2)MAV_CMD_DO_MOUNT_CONTROL

l)gimbal_CONTROL_MODE_ANGLE

ll)gimbal_CONTROL_MODE_SPEED

lll)gimbal_CONTROL_MODE_ANGLE_REF_FRAME

3)MAV_CMD_REQUEST_CAMERA_INFORMATION

4)MAV_CMD_REQUEST_CAMERA_SETTINGS

5)MAV_CMD_SET_CAMERA_MODE

6)MAV_CMD_REQUEST_STORAGE_INFORMATION

7)MAV_CMD_STORAGE_FORMAT

8)MAV_CMD_REQUEST_CAMERA_CAPTURE_STATUS

9)MAV_CMD_IMAGE_START_CAPTURE

10)MAV_CMD_VIDEO_START_CAPTURE

11)MAV_CMD_VIDEO_STOP_CAPTURE

12)MAV_CMD_REQUEST_MESSAGE

13)MAV_CMD_SET_CAMERA_ZOOM

C)MAVLINK_MSG_ID_ATTITUDE

D)MAVLINK_MSG_ID_GLOBAL_POSITION_INT

E)MAVLINK_MSG_ID_RC_CHANNELS

F)MOUNT_ORIENTATION

G)MAV_CMD_DO_GIMBAL_MANAGER_ATTITUDE

H)MAV_CMD_DO_GIMBAL_MANAGER_TRACK_POINT (1001)

I)MAV_CMD_DO_GIMBAL_MANAGER_TRACK_RECTANGLE (1002)

A) control the gimbal angle,

SP1: MAVLINK_MSG_ID_COMMAND_LONG-->MAV_CMD_DO_MOUNT_CONFIGURE

 s_command_long.param1 = MAV_MOUNT_MODE_MAVLINK_TARGETING

s_command_long.param5 = 2;

 SP2:MAV_CMD_DO_MOUNT_CONTROL

s_command_long.param7 = MAV_MOUNT_MODE_MAVLINK_TARGETING

 angle s_command_long.param1 = angle

B) camera zoom,

MAV_CMD_SET_CAMERA_ZOOM

1: Zoom Type Zoom type CAMERA_ZOOM_TYPE

2: Zoom Value Zoom value. The range of valid values depend on the zoom type.

(note: CAMERA_ZOOM_TYPE

[Enum] Zoom types for MAV_CMD_SET_CAMERA_ZOOM

Value Field Name Description

- 0 *ZOOM_TYPE_STEP* Zoom one step increment (-1 for wide, 1 for tele)
- 1 *ZOOM_TYPE_CONTINUOUS* Continuous zoom up/down until stopped (-1 for wide, 1 for tele, 0 to stop zooming)
- 2 *ZOOM_TYPE_RANGE* Zoom value as proportion of full camera range (a value between 0.0 and 100.0)
- 3 *ZOOM_TYPE_FOCAL_LENGTH* Zoom value/variable focal length in millimetres. Note that there is no message to get the valid zoom range of the camera, so this can type can only be used for cameras where the zoom range is known (implying that this cannot reliably be used in a GCS for an arbitrary camera))

C)photo taking

MAV_CMD_IMAGE_START_CAPTURE

D)recording

MAV_CMD_VIDEO_START_CAPTURE

MAV_CMD_VIDEO_STOP_CAPTURE

E)receive gimbal angle information,

MOUNT_ORIENTATION (send any data for feedback)

Or MAV_CMD_REQUEST_MESSAGE

F)Set photo and video mode

---MAV_CMD_SET_CAMERA_MODE

G)Get photo and video mode information.

MAV_CMD_REQUEST_MESSAGE

H)---CAMERA_MODE

CAMERA_MODE

[Enum] Camera Modes.

Value	Field Name	Description
0	CAMERA_MODE_IMAGE	Camera is in image/photo capture mode.
1	CAMERA_MODE_VIDEO	Camera is in video capture mode.
2	CAMERA_MODE_IMAGE_SURVEY	Camera is in image survey capture mode. It allows for camera controller to do specific settings for surveys.

Camera zoom:

Zoom in

Zoom out

Zoom stop

Take picture:

FD 20 00 00 16 FF 00 4C 00 00 00 00 00 00 00 00 00 00 00 80 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D0 07 01 64 96 AE

COMMAND_LONG {target_system: 1, target_component: 100, command: 2000, confirmation: 0, param1: 0.0, param2: 0.0, param3: 1.0, param4: 0.0, param5: 0.0, param6: 0.0, param7: 0.0}

Start record

Stop record