-------------------------------------------------------------------------------

快拆MCU软件通讯功能说明：

1. MCU通过CAN接口接收指令，把相机指令通过UART2发给相机，把云台指令通过UART3发到云台控制器。

CAN波特率100K。

UART2波特率9600，8为数据位，1位停止位，无奇偶校验，无流控制。

UART3波特率500k，8为数据位，1位停止位，无奇偶校验，无流控制。

1. CAN接收报文：

接收ID：0x200，接收报文类型均为标准数据帧。

接收报文格式如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CON\_IDX | DATA1 | DATA2 | DATA… |  |  |  |  |

CON\_IDX：

Bit0~3：报文序号，每收到一次ID为0x200的报文，此项都应加一，顺序排列，循环使用，如果序列号顺序出错，将发出错报告帧，详见CAN发送报文部分。

Bit4：需要确认报文使能位；

0x1：接到报文后，需发报文确认，确认报文详见CAN发送报文部分；

0x0：接到报文后，不需发报文确认。

Bit5：为0，保留。

Bit6~7：为报文数据层控制位；（保留，暂没用到）

0x00：单帧数据报文，即数据在单帧（在本帧内）内结束；

0x01：多帧数据报文的起始帧；

0x11：多帧数据报文的中间帧；

0x10：多帧数据报文的结束帧。

数据层数据长度大于7时，通过CON\_IDX的数据层控制位进行分解和组合。

DATA1：数据报文识别

0xFF：相机指令直接传输，即将接收到的DATA数据直接转发；

0xFE：相机指令，将0xFE 用0xFF 01 04 替换，衔接其他DATA数，再发送给相机， 详见后续用到此标识处；

0xFA：云台yaw pitch指令，转换为MAVLink控制姿态指令发送至云台，详见后续 用到此标识部分。

1. CAN发送报文：

发送ID：0x201，发送报文类型均为标准数据帧。

格式为四字节：

|  |  |  |  |
| --- | --- | --- | --- |
| INDEX | ACK\_CODE | ACK\_D1 | ACK\_D2 |

INDEX Bit0~3：报文序号，循环加1；Bit4~7,保留。

ACK \_CODE: 返回码，返回码Bit7为1，则为出错报告报文；

0x01：确认报文，ACK\_D1为接收报文的序号，即CON\_IDX的Bit0~4位，ACK\_D2 位0。当收到报文的需发确认报文使能位为1时，则发送此报文作为确认； 如接收的报文出错，则发出错报告报文当作确认。

0x81：出错报告报文，接收报文序号顺序错误，ACK\_D1为接到报文序号，ACK\_D1 为接到的上一帧报文序号。

0x82：解析错误，ACK\_D1为接到报文序号，ACK\_D2位0。

1. 相机指令（FF 01 00 00 00 00 01 //停止）

CAN接收8字节：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CON\_IDX | 0xFF | 0x01 | 0x00 | 0x00 | 0x00 | 0x00 | 0x01 |

UART2发送7字节：

FF 01 00 00 00 00 01

1. 相机指令（FF 01 01 00 00 00 02 //聚焦近）

CAN接收8字节： CON\_IDX FF 01 01 00 00 00 02

UART2发送7字节：

FF 01 01 00 00 00 02

1. 相机指令（FF 01 00 80 00 00 81 //聚焦远） 同上
2. 相机指令 拍照 FF 01 00 07 00 E1 E9 同上
3. 相机指令 录像开启 FF 01 00 07 00 E2 EA 同上
4. 相机指令 录像关闭 FF 01 00 07 00 E3 EB 同上
5. 相机指令 初始化程序 FF 01 00 07 00 DF E7 同上
6. 相机指令 镜头位置输入Zoom（1x，2x，3x，4x……30x）参见ZoomData表

CAN接收8字节： CON\_IDX FE 47 0p 0q 0r 0s FF

UART2发送9字节：（0xFE 用FF 01 04替换）

FF 01 04 47 0p 0q 0r 0s FF

1x 2x… 对应不同p q r s，查ZoomData表

注：[原文广播不是FF，而是88（88 01 04 47 0p 0q 0r 0s FF），头字节Bit3 为广播标志]

注：[原文：Zoom Direct 8x 01 04 47 0p 0q 0r 0s FF ]

1. 相机指令 曝光补偿 exposurecompensation相关设置

注：【

CAM\_ExpComp

On 8x 01 04 3E 02 FF Exposure Compensation On/Off

Off 8x 01 04 3E 03 FF

Reset 8x 01 04 0E 00 FF Exposure Compensation Amount Setting

Up 8x 01 04 0E 02 FF

Down 8x 01 04 0E 03 FF

Direct 8x 01 04 4E 00 00 0p 0q FF pq: ExpComp Position

】

曝光补偿 on：

CAN接收5字节： CON\_IDX FE 3E 02 FF

UART2发送6字节：

FF 01 04 3E 02 FF

曝光补偿 off：

CAN接收5字节： CON\_IDX FE 3E 03 FF

UART2发送6字节：

FF 01 04 3E 03 FF

曝光补偿 Reset：

CAN接收5字节： CON\_IDX FE 0E 00 FF

UART2发送6字节：

FF 01 04 0E 00 FF

曝光补偿 Up：

CAN接收5字节： CON\_IDX FE 0E 02 FF

UART2发送6字节：

FF 01 04 0E 02 FF

曝光补偿 Down：

CAN接收5字节： CON\_IDX FE 0E 03 FF

UART2发送6字节：

FF 01 04 0E 03 FF

曝光补偿 Direct：

CAN接收8字节： CON\_IDX FE 4E 00 00 0p 0q FF

UART2发送9字节：

FF 01 04 4E 00 00 0p 0q FF

1. 相机指令 分辨率&帧数设置

注：

【Register Setting Command: 8x 01 04 24 mm 0p 0q FF

mm: Register No. pq: Register Value (=00 to FF)

Register No. 72 （0x72）

Value：

01 (Iinitial Setting) 1080i/59.94 Reflected after camera reset

02 1080i/60

04 1080i/50

06 1080p/29.97

07 1080p/30

08 1080p/25

09 720p/59.94

0A 720p/60

0B Reserved

0C 720p/50

0D Reserved

0E 720p/29.97

0F 720p/30

10 Reserved

11 720p/25

12 Reserved

13 1080p/59.94

14 1080p/50

15 1080p/60

】

CAN接收7字节： CON\_IDX FE 24 72 0p 0q FF

UART2发送8字节：

FF 01 04 24 72 0p 0q FF

1. 相机指令 云台指令，通过UART3发至云台。

俯仰控制&回中

航向控制&回中

CAN接收7字节： CON\_IDX FA pp qq mm nn xy

ppqq: yaw定点小数；

mmnn: pitch定点小数；

x：yaw mode，4 bits；

y：pitch mode，4 bits；

注：需提供定点小数格式以便转换为浮点，或提供转换函数。

UART2发送10+20+2=32字节：

MAVLink报文：

10字节头 + float yaw\_channel float pitch\_channel float roll\_channel float drones\_yawvelocity\_desire uint8\_t priority uint8\_t yaw\_mode uint8\_t pitch\_mode uint8\_t roll\_mode + CRC

Roll参数填缺省值。

注：

【MAVLink：

void gimbal\_control\_standard\_send(mavlink\_channel\_t chan, uint8\_t priority, uint8\_t yaw\_mode, uint8\_t pitch\_mode, uint8\_t roll\_mode, float yaw\_channel, float pitch\_channel, float roll\_channel, float drones\_yawvelocity\_desire)；

gimbal\_control\_standard\_send(0, 20, 3, 3, 4, \_yaw, \_pitch, 0, 0);

gimbal\_control\_standard\_send(0, 20, yaw\_mode, pitch\_mode, 4, \_yaw, \_pitch, 0, 0);

Header：

buf[0] = MAVLINK\_STX;

buf[1] = length;

buf[2] = incompat\_flags;

buf[3] = 0; // compat\_flags

buf[4] = status->current\_tx\_seq;

buf[5] = 1;

buf[6] = 154;

buf[7] = msgid & 0xFF;

buf[8] = (msgid >> 8) & 0xFF;

buf[9] = (msgid >> 16) & 0xFF;

MAVPACKED(

typedef struct \_\_mavlink\_gimbal\_control\_standard\_t {

float yaw\_channel; /\*< yaw channel value,if yaw mode is angle ,this value is angle.if yaw mode is velocity,this value is velocity..\*/

float pitch\_channel; /\*< pitch channel value,if pitch mode is angle ,this value is angle.if pitch mode is velocity,this value is velocity..\*/

float roll\_channel; /\*< roll channel value,if roll mode is angle ,this value is angle.if roll mode is velocity,this value is velocity..\*/

float drones\_yawvelocity\_desire; /\*< drone's yaw desired anguler velocity.\*/

uint8\_t priority; /\*< the priority of this control packet.range is 1~253,1 is the highest and 253 is lowest;self-control is 1,the packet send by GIMBAL\_CONTROL is 10,and the packet send by GIMBAL\_DEBUGDATA is 20,the others number belong to user. \*/

uint8\_t yaw\_mode; /\*< yaw mode value,1 is follow\_head,2 is follow\_angle,3 is follow\_velocity;4 is global\_angle,5 is global\_velocity.\*/

uint8\_t pitch\_mode; /\*< pitch mode value,1 is follow\_head,2 is follow\_angle,3 is follow\_velocity;4 is global\_angle,5 is global\_velocity.\*/

uint8\_t roll\_mode; /\*< roll mode value,1 is follow\_head,2 is follow\_angle,3 is follow\_velocity;4 is global\_angle,5 is global\_velocity.\*/

}) mavlink\_gimbal\_control\_standard\_t;

MAVLINK\_SEND\_UART\_BYTES(chan, buf, header\_len+1);

MAVLINK\_SEND\_UART\_BYTES(chan, (uint8\_t\*)packet, length);

MAVLINK\_SEND\_UART\_BYTES(chan, ck, 2);





】