上下位机通讯协议

数据格式:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 8bit | 32bit | 8bit | ... | 8Bit | 8bit | 8bit |
| 包头 | 包长 | 包类型 | 数据 | 是否应答 | crc | 包尾 |
|  | CRC校验范围 | | | |  |  |
| 包长 | | | | | | |

1. #ifndef \_\_PROTOCOL\_H
2. #define \_\_PROTOCOL\_H
4. #define PROTOCOL\_HEAD           (0x55)   包头
5. #define PROTOCOL\_TAIL          (0xaa)  包尾
6. 数据包的类型枚举
7. **typedef** **enum** \_protocol\_type {
8. E\_PROTOCOL\_NULL\_CMD = 0 没有命令包
9. E\_PROTOCOL\_MOVING\_CMD ,   运控控制包
10. E\_PROTOCOL\_HEADING\_SYCN,   航向同步包 当应答位为1时表示停止发送
11. E\_PROTOCOL\_POSITION\_SYCN,   位置同步包 当应答位为1时表示停止发送
12. E\_PROTOCOL\_LOCATION\_DATA,   实时定位包 当应答位为1时表示停止发送
13. E\_PROTOCOL\_MACHINE\_STATE,   机器状态包
14. E\_PROTOCOL\_ANSWER\_CMD, 回应接收包 55 09 00 00 00 06 00 3C aa
15. E\_PROTOCOL\_REMOTE\_CMD, 上位机遥控包 数据中的命令格式与运动控制包相同
16. E\_PROTOCOL\_SIZE,   协议数量
17. } E\_protocol\_type;
18. 定位包开关
19. **typedef** **struct** \_location\_cmd{
20. uint8\_t start\_or\_stop; 上位机下发跟定位相关包,数据段位这个结构体,1为启动,0为停止(航向同步,位置同步,实时定位)
21. }ST\_location\_cmd;
22. 运动控制命令类型枚举
23. **typedef** **enum** \_protocol\_moving\_cmd{
24. E\_PROTOCOL\_MOVING\_STOP = 0,   停止命令
25. E\_PROTOCOL\_MOVING\_BRAKE,   刹车命令
26. E\_PROTOCOL\_MOVING\_MOVING\_STRAIGHT,   走直线命令,v>0前进,v<0后退
27. E\_PROTOCOL\_TURN\_TO,   转弯命令,w>0,w<0,
28. **E\_REMOTE\_PROTOCOL\_STOP,**
29. **E\_REMOTE\_PROTOCOL\_BRAKE,**
30. **E\_REMOTE\_PROTOCOL\_MOVING\_LINE,**
31. **E\_REMOTE\_PROTOCOL\_TURN,**
32. }E\_protocool\_moving\_cmd;
34. 定位状态枚举:有效,无效
35. **typedef enum \_gps\_location\_state{**
36. **E\_GPS\_STATE\_V =0,**
37. **E\_GPS\_STATE\_A,**
38. **}E\_gps\_location\_state;**  定位模式枚举: rtk模式,float模式..
39. 可能值：N=无固定，E=估计/死亡推算修正，A=自主GNSS固定，D=差分GNSS固定，F=RTK浮动，R=RTK固定
40. **typedef enum \_gps\_location\_mode{**
41. **E\_GPS\_MODE\_N = 0,**
42. **E\_GPS\_MODE\_E,**
43. **E\_GPS\_MODE\_A,**
44. **E\_GPS\_MODE\_D,**
45. **E\_GPS\_MODE\_F,**
46. **E\_GPS\_MODE\_RTK,**
47. **}E\_gps\_location\_mode;**
48. 维度:南纬,北纬
49. **typedef** **enum** \_gps\_latitude{
50. E\_GPS\_N= 0,
51. E\_GPS\_S,
52. }E\_gps\_latitude;
53. 经度:东经,西经
54. **typedef** **enum** \_gps\_longitude{
55. E\_GPS\_E= 0,
56. E\_GPS\_W,
57. }E\_gps\_longitude;
59. #pragma pack (1)
60. 运动控制数据结构体:
61. **typedef** **struct** \_moving\_cmd{
62. uint8\_t id\_num;
63. uint8\_t  e\_protocool\_moving\_cmd;
64. Int16\_t v;
65. float w;
66. float s\_or\_angle;
67. }ST\_moving\_cmd;
68. Typedef struct \_answer\_cmd{
69. uint8\_t ans;//--0 , 1
70. }ST\_answer\_cmd,\*ST\_answer\_cmd\_p;
71. Ekf定位系统数据结构体
72. **typedef** **struct** \_ekf\_location{
73. uint32\_t base\_time;
74. uint32\_t imu\_time;
75. **double** pitch;
76. **double** roll;
77. **double** yaw;
78. **double** acc\_y;
79. **double** gyo\_x;
80. **double** gyo\_y;
81. **double** gyo\_z;
82. uint32\_t gps\_time;
83. uint8\_t e\_gps\_location\_state;
84. **double** latitude;
85. uint8\_t e\_gps\_latitude ;
86. **double** longitude;
87. Uint8\_t e\_gps\_latitude;
88. **float** gps\_speed;
89. Uint8\_t E\_gps\_location\_mode;
90. uint16\_t gps\_rms;
91. uint32\_t odo\_time;
92. int32\_t odo\_speed;
93. }ST\_ekf\_location;
95. 通信数据包结构
96. **typedef** **struct** \_protpcol\_info {
97. uint8\_t head;
98. uint32\_t package\_len;
99. Uint8\_t e\_protocol\_type ;
100. uint8\_t\* p\_data;
101. uint8\_t crc;
102. uint8\_t tail1;
103. uint8\_t tail2;
104. } ST\_protocol\_info;
105. #pragma pack (0)
106. 通讯模块对外接口
107. **typedef** **struct** \_protocol {
108. 初始化
109. uint8\_t (\*init)(**void**);
110. 发送数据包
111. 参数: e\_protocol\_type 数据包类型
112. 参数: p\_data 数据
113. 参数: data\_len 数据长度
114. uint8\_t (\*send)(E\_protocol\_type e\_protocol\_type, **const** **void**\* p\_data, uint32\_t data\_len);
115. uint8\_t (\*parse\_protocol)(**void**);
116. uint8\_t (\*deinit)(**void**);
117. } ST\_protocol;
118. extern ST\_protocol protocol;
119. #endif

通讯模块功能描述:

* 1. 跟上位机行通讯,作为下位机各个模块与上位机通讯的接口.
  2. 对要发送给上位机的数据打包,自动添加包头包尾,校验等数据包特征.例:

protocol.send(E\_PROTOCOL\_LOCATION\_DATA,ST\_ekf\_location\*st\_ekf\_location\_data, sizeof(ST\_ekf\_location))

* 1. 解析上位机下发的数据,命令,根据命令执行想用模块的功能.(以命令队列的形式,按顺序执行命令)
  2. 上位机下发命令:运动控制,航向同步,位置同步等包需要回复.其中,运动控制有参数,航向同步,位置同步无参数(没有数据),下位机结束发送时,最后一个数据包回复位设为1,如果未回复,重发数次后报错.
  3. ,上位机下发的命令需要下位机回应,如下位机未回应,则持续发送命令数次,后报错.下位机上报数据结束,并等待上位机回应,如未回应则持续发送数次后,报错.

上位机

下位机

上报下位机传感器等一些信息（10ms发送一次）

接受到正确命令，加入命令队列

解析命令

为了缓解下位机压力，上位机发送控制命令后，如果接收到下位机回应，就停止发送，除非有新的命令产生

下位机检测命令队列，有数据就继续执行 ，没数据，就保持当前状态或停止

接收到正确回应，停止发送

回应上位机

10ms发送一次

下发控制命令