**机器人间通信**

测试前确认双方都已经连好服务器

在referee.h 里约定好数据格式 图中为一个例子

typedef \_\_packed struct

{

    //uint8\_t data[112];

    uint8\_t sentry\_chassis\_state;

    uint8\_t sentry\_gimbal\_state;

    uint8\_t sentry\_fire\_state;

} ext\_robot\_interactive\_data\_t; //机器人间通信数据

发送方： 在ui.c里面 改一下原来的使用如下发送函数 发送ID和接受ID使用对应的（查协议 或者ui.c里应该已经写了）

自己写下SentryInteractiveSet（）函数用于更改发送数据

最后在freertos里面相应线程调用一下，不要把发送频率调太高 10hz足够

void Robot\_Interactive(uint16\_t send\_ID, uint16\_t recv\_ID)

{

    int index=0;

    uint8\_t packed\_robot\_inter\_data[18];

    uint16\_t cmd\_id = 0x0301;

    frame\_header\_struct\_t front\_sight\_header = {(uint8\_t)0xA5,9,(uint8\_t)1,0};

    append\_CRC8\_check\_sum((uint8\_t\*)&front\_sight\_header,5);

    ext\_student\_interactive\_header\_data\_t front\_sight\_data\_header = {0x0200,send\_ID,recv\_ID};

    ext\_robot\_interactive\_data\_t robot\_inter\_data;

    memset(&robot\_inter\_data,0,sizeof(ext\_robot\_interactive\_data\_t));

    SentryInteractiveSet(&robot\_inter\_data);

    memcpy(packed\_robot\_inter\_data+index,&front\_sight\_header,sizeof(front\_sight\_header));

    index+=sizeof(front\_sight\_header);

    memcpy(packed\_robot\_inter\_data+index,(uint8\_t\*)&cmd\_id,sizeof(cmd\_id));

    index+=sizeof(cmd\_id);

    memcpy(packed\_robot\_inter\_data+index,&front\_sight\_data\_header,sizeof(front\_sight\_data\_header));

    index+=sizeof(front\_sight\_data\_header);

    memcpy(packed\_robot\_inter\_data+index,&robot\_inter\_data,sizeof(robot\_inter\_data));

    append\_CRC16\_check\_sum(packed\_robot\_inter\_data,sizeof(packed\_robot\_inter\_data));

    HAL\_UART\_Transmit(&huart6,packed\_robot\_inter\_data,sizeof(packed\_robot\_inter\_data),20);

}

接受方：确认以下在referee\_solve.c里面有没有相应解析（应该已经写了）

开debug确认下收到的数据是否有误

  case STUDENT\_INTERACTIVE\_DATA\_CMD\_ID:

    memcpy(&student\_inter\_header\_t, frame + index, sizeof(student\_inter\_header\_t));

    memcpy(&robot\_inter\_data\_t, frame+index+sizeof(student\_inter\_header\_t), sizeof(robot\_inter\_data\_t));

    break;

  default:

    break;