**SHRD100\_测量信息驱动详细设计**

|  |  |
| --- | --- |
| 拟 制 |  |
| 审 核 |  |
| 会签 |  |
| 批 准 |  |

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **修订版本** | **日期** | **作者** | **修改描述** | **备注** |
| V1.0 | 2023.09.18 | 贾玮 | 初始版本 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 引言 3](#_Toc145946459)

[1.1 目的 3](#_Toc145946460)

[1.2 范围 3](#_Toc145946461)

[1.3 缩略语定义 3](#_Toc145946462)

[1.4 参考资料 3](#_Toc145946463)

[2 需求概述 3](#_Toc145946464)

[2.1 功能需求 4](#_Toc145946465)

[2.1.1 Tracer与UAV的通信使用UsbTypeC接口 4](#_Toc145946466)

[2.1.2 Tracer驱动提供统一接口查询角度和飞行姿态信息 4](#_Toc145946467)

[2.2 接口需求 4](#_Toc145946468)

[2.3 性能需求 4](#_Toc145946469)

[2.4 边界需求 4](#_Toc145946470)

[3 软件设计说明 5](#_Toc145946471)

[3.1 设计示意图 5](#_Toc145946472)

[3.2 驱动初始化 5](#_Toc145946473)

[3.3 接受UAV消息处理流程 7](#_Toc145946474)

[3.4 Tracer接受从UAV收到Alink消息 (0xE5)处理 7](#_Toc145946475)

[3.5 发送测量信息到UAV 8](#_Toc145946476)

[4 Issues解答 8](#_Toc145946477)

[4.1 问题1 8](#_Toc145946478)

[4.2 问题1 8](#_Toc145946479)

[5 附件 9](#_Toc145946480)

# 引言

## 目的

本文为“SHRD100 Tracer与UAV 之间飞行姿态,侦测目标的测量值等数据交互的驱动软件详细设计”,主要用于指导软件编码人员完成Tracer解析UAV数据, 并提供驱动接口给算法模块,供项目组开发人员和软件维护人员阅读。

## 范围

本文档只限于塞防科技项目组研发、测试、产品以及项目相关人员作为内部信息对齐使用，未经公司批准以及书面授权不允许任何人以任何形式对本文档复制、传播、改动。

## 缩略语定义

|  |  |  |
| --- | --- | --- |
| **缩略语** | **全称** | **描述** |
| UAV | Unmanned Aerial Vehicle | 无人驾驶空中飞行器 |
|  |  |  |

## 参考资料

|  |  |
| --- | --- |
| **名称** | **版本** |
| SHRD100\_系统设计说明书 |  |
| SHRD100嵌入式软件设计说明V1.3.docx |  |
| 机载电侦产品软件系统设计及处理流程 20230914.docx |  |
| tracer与C2和上位机通信协议V1.0.12.docx |  |
|  |  |

# 需求概述

SHRD100 Tracer产品新增无人机机载版本，原有本地陀螺仪采集数据不满足机载的复杂环境，需要由无人机把飞行姿态数据通过UsbTypeC发送给Tracer，Tracer的驱动提供接口供算法模块使用，另一方面Tracer对侦测目标的测量数据通过UsbTypeC发送给UAV。

## 功能需求

### Tracer与UAV的通信使用UsbTypeC接口

通信协议使用MavAlink，协议内容可参考tracer与C2和上位机通信协议V1.0.12.docx。 通信数据采用protobuf格式定义，UavTracerMsg.proto文件见附件1。

### Tracer驱动提供统一接口查询角度和飞行姿态信息

原有Tracer通过TracerDeviceHal从本地硬件获取角度信息数据，新增机载版本通过UsbTypeC从UAV获取数据方式，驱动设计目标是提供统一的接口，屏蔽数据获取方式，使算法应用无需关心数据来源。

## 接口需求

通信协议使用MavAlink，协议内容可参考tracer与C2和上位机通信协议V1.0.12.docx。 通信数据采用protobuf格式定义，UavTracerMsg.proto文件见附件1。

## 性能需求

|  |  |
| --- | --- |
| 序号 | 性能需求 |
| 1 | UAV每120ms会发送1次飞行姿态信息数据，Tracer算法获取驱动数据并计算至少小于120ms。驱动设计目标从UsbTypeC收到数据并存贮的时间小于5ms，提供算法查询数据接口的时间消耗小于5ms |
| 2 |  |

## 边界需求

无

# 软件设计说明

## 设计示意图

**算法**

算法查询飞行姿态信息

Tracer向UAV发送测量信息

\_angleData: AngleInfoPB

\_aapsData: FlyStatusPB

ParseFromArray ()

SerializeToArray ()

**飞行数据Cache**

驱动设计

\_data: AAPSInfo

SendMeasureInfoMsgToUav()

RecvFlayStatusFromUav()

**TracerDeviceHal**

查询数据

发送测量数据

接收回调函数,存储数据

**网络侧(UsbTypeC)接口**

UsbTypeC，消息格式AlinkMsg

**UAV(UsbTypeC)接口**

## 驱动初始化

由于Tracer和C2使用UsbTypeC通信， 为了尽快支持机载版本发布，Tracer机载版本直接重用UsbTypeC驱动使用c2\_network提供的函数接口。当前版本UsbTypeC接口不会同时接C2和UAV， 理论上重用c2\_network不会带来负面影响。

1. TracerCommonHal::Init()
2. c2\_network\_init()设置eth\_link\_user\_t为c2\_network\_create
3. c2\_network\_create()需判断是否机载版本，如是机载版本发送一次心跳报文EAP\_ALINK\_RPT\_HEARTBEAT(0xEF)

eth\_link\_cbk\_t\* c2\_network\_create( eth\_link\_user\_t \*psUser )

{

c2\_network\_t \*psNetwork = (c2\_network\_t\*)malloc(sizeof(c2\_network\_t));

printf( "[%s %d]\r\n", \_\_func\_\_, \_\_LINE\_\_ );

if( psNetwork )

{

memset( psNetwork, 0, sizeof(c2\_network\_t) );

if (机载版本)

{

alink\_connect\_send( &psNetwork->sSendAlinkHeartBeat, EAP\_ALINK\_RPT\_HEARTBEAT, ALINK\_DEV\_ID\_UVA, psUser, c2\_network\_send);

}

else

{

//c2 沿用处理逻辑

}

alink\_connect\_port( &psNetwork->sAlinkSocket, NULL );

psNetwork->psUser = psUser;

psNetwork->sCbk.psHandle = psNetwork;

psNetwork->sCbk.pv\_cbk\_connect = c2\_network\_cbk\_connect;

psNetwork->sCbk.pv\_cbk\_tcp\_Isok = c2\_network\_cbk\_tcp\_Isok;

psNetwork->sCbk.pv\_cbk\_data = c2\_network\_cbk\_data;

psNetwork->sCbk.pv\_cbk\_disconnect = c2\_network\_cbk\_disconnect;

psNetwork->sCbk.pv\_cbk\_delete = c2\_network\_cbk\_delete;

return &(psNetwork->sCbk);

}

return NULL;

}

新增类型定义：

ALINK\_DEV\_ID\_UAV = (8)

ALINK\_PORT\_ID\_TCP\_UAV = (4)

## 接受UAV消息处理流程

1.c2\_network\_cbk\_data()检查消息header和CRC

2. send\_alinkmsg\_to\_eap()

3. SendAsynMsg()发送EAP\_CMD\_ALINK

4. AlinkCmdproc::\_OnAlinkCmd，destId需等于本设备ID(Tracer)

5. \_ProcAlinkCmd 调用在void SysBase::\_RegisterAlinkCmd()注册的回调函数进行处理

void SysBase::\_RegisterAlinkCmd()

{

register\_alink\_cmd(EAP\_ALINK\_CMD\_TRACER\_UAV\_MESSAGE, Alink\_TracerUavMsgProc\_Func);

}

## Tracer接受从UAV收到Alink消息 (0xE5)处理

Alink message的payload是TracerUavMessagePB，根据TracerUavMessagePB的msgType属性，调用不同处理函数

Alink\_TracerUavMsgProc\_Func()

{

if (TRACERUAVMSG\_WORK\_MODE\_TYPE == msgType)

{

使用SendAsynData发送EAP\_CMD\_SET\_WORK\_MODE

设置EapGetSys().SetWorkMode

}

else if (TRACERUAVMSG\_FLY\_STATUS\_TYPE == msgType)

{

设置飞行姿态数据到HAL，

halMgr->OnSet(EAP\_DRVCODE\_SET\_UAV\_STATUS, inData);

}

else

{

// do nothing

}

}

## 发送测量信息到UAV

注册发送MeasureInfoToUav函数

TracerDeviceHal::TracerDeviceHal(HalMgr\* halMgr) : HalBase(halMgr)

{

\_Register(EAP\_DRVCODE\_RPT\_UAV\_MEASURE\_INFO, (Hal\_Func)& TracerDeviceHal::\_SendMeasureInfoMsgToUav);

}

算法调用TracerUavMessagePB把消息序列化后调用halMgr->OnSet(EAP\_DRVCODE\_RPT\_UAV\_MEASURE\_INFO, inData); 发送测量信息到Uav

# Issues解答

## 问题1

## 问题1

# 附件

1. protobuf 接口定义：