第 2 卷◆第 4 期◆版本 1.0◆2024 年
文章类型：论文|刊号（ISSN）：2972-4236(P) / 2972-4244(O)

多点定位数据解码及格式转换软件的设计与实现

李建宏　龙滨
民航重庆空管分局
DOI:10.12238/acair.v2i4.10742

[摘　要] ASTERIX Cat010由Euro Control组织开发,通常用于传输场监雷达、多点定位、ADS-B数据的传输。在重庆空管,主要用于多点定位信号传输。ASTERIX Cat010原始数据不能直接阅读,需要进行解码。本文介绍了一种多点定位数据解码及格式转换软件的实现方法,该软件能够将复杂的原始数据转化为易于理解的格式,从而满足用户单位的信息需求。通过该软件,用户可以实时获取场面航班信息,提升安全保障水平,优化运营流程,并提高服务质量和效率。
[关键词] 多点定位；ASTERIX; CAT010；解码；格式转换；QT Creator；C++
中图分类号：P228.1　文献标识码：A

Design and implementation of multi-point location data decoding and format conversion software

Jianhong Li　Bin Long
Chongqing Air Traffic Control Branch of CAAC

[Abstract] ASTERIX Cat010 is developed by Euro Control organization, and is usually used for the transmission of field surveillance radar, multi-point location, and ADS-B data. In Chongqing air traffic control, mainly used for multi-point location signal transmission. ASTERIX Cat010 The raw data cannot be read directly and needs to be decoded. This paper introduces an implementation method of multipoint location data decoding and format conversion software, which can convert complex raw data into an easily understandable format to meet the information needs of user units. Through the software, users can obtain real-time scene flight information, improve the level of security, optimize the operation process, and improve the service quality and efficiency.
[Key words] multi-point positioning; ASTERIX; CAT010; decoding; format conversion; QT Creator; C++

随着航空业的快速发展,机场场面监控成为确保飞行安全和提高运营效率的关键环节。多点定位系统(Multilateration System)作为一种先进的场面监控技术,能够实时提供航班的位置、速度等信息。然而,场监雷达和多点定位数据通常采用ASTERIX Cat010等复杂的数据编码格式,这对于非专业人士来说难以理解和使用。因此,开发一款能够将多点定位数据解码并转换为易于理解和使用的格式的软件具有重要意义。

本人在QT Creator编程软件中利用C++语言进行具有图形界面的CAT010多点定位数据的解码及格式转换软件开发。实现了以下4个主要功能：

(1)多点定位原始数据的UDP实时录取；(2)多点定位数据解码,包括Asterix Cat010数据格式研究以及在软件中的解码实现；(3)根据用户需求,进行个性化数据定制输出；(4)可视化软件界面实现软件解码及数据输出的状态监控及参数配置。

1 ASTERIX Cat010基本数据结构分析

ASTERIX Cat010格式由CAT(Category,类别)、LEN(Length,长度)和数据记录三部分组成。数据记录进一步由FSPEC(Field Specification,字段说明)和各数据字段组成。

CAT：表示数据报文的类别,ASTERIX Cat010即为多点定位数据的报文类别。

LEN：表示数据报文的长度,包括CAT、LEN、FSPEC和各数据字段的总长度。

FSPEC：字段说明部分,以位为单位标识每一个字段是否存在。FSPEC的存在使得数据报文具有高度的灵活性和可扩展性,能够根据不同的需求包含不同的数据字段。

各数据字段：包含具体的飞行器数据,如航班号、位置、速度以及目标是否在地面、目标类型识别(行李、消防、公交、燃油)等信息。

其结构如下表所示。因篇幅原因只介绍前两个常用字段。

字段参考编码FRN	数据项Data Item	信息Information	字节长度Length in Octets
1	I010/010	数据源标识	2
2	I010/000	信息类型	1
3	I010/020	目标报告描述符	1+
4	I010/140	当日时间	3
5	I010/041	WGS84 纬度和经度	8
6	I010/040	极坐标测量位置	4
7	I010/042	笛卡尔坐标位置	4
FX	-	字段扩展标识位	-
8	I010/200	极坐标下的计算航迹速度	4
9	I010/202	笛卡尔坐标系下的计算航迹速度	4
10	I010/161	航迹号	2
11	I010/170	航迹状态	1+
12	I010/060	Mode-3 A模式二次码	2
13	I010/220	目标地址	3
14	I010/245	目标识别	7
FX	-	字段扩展标识位	-
.....	.....	.....	.....

2 多点定位原始数据的UDP实时录取

为实现多点定位数据的实时处理,软件采用UDP协议进行数据传输。通过创建UDP套接字并绑定到指定端口,软件不断监听来自多点定位系统的数据报文,接收后进行解码和格式转换。实现代码(部分)如下。

2.1 UPD连接初始化

```
void MainWindow::initUDP(QString ip,int port)
{
connect(udpSocket,SIGNAL(readyRead()),this,SLOT(dataReceived()));
bool result=udpSocket->bind(QHostAddress(ip),port);
if(!result)
{
QMessageBox::information(this,tr("error"),tr("udp socket create error!"));
}
}
```

2.2多点定位系统原始数据获取

```
void MainWindow::dataReceived()//接收原始数据
{
try
{
while(udpSocket->hasPendingDatagrams());//判断是否有数据
{
QByteArray datagram;
datagram.resize(udpSocket->pendingDatagramSize());
udpSocket->readDatagram(datagram.data(),datagram.size());
BYTE dataArray[datagram.size()];//初始化数组
BYTE*p =(BYTE *)datagram.data();//获取数据报的首地址
memcpy(dataArray,p,datagram.size());//复制数据到dataArray
if(p[0]=0x0A and dataReceiverOn)
{
qDebug()<<"This array iscat.010";
asterixCat010.dataParser(dataArray);
MainWindow::printCat010();//打印数据
counter++;
}
else
{
qDebug()<<"This array is not cat.010";
}
}
}
}
```

3 多点定位数据解码

数据解码是软件的核心功能之一。解码过程包括解析CAT和LEN字段、解析FSPEC字段以及解析数据字段。软件采用C++的位运算和结构体特性,高效地解析数据报文。实现代码(部分)如下。

3.1 FSPEC字节组成定义

```
struct FspecData
{
BYTE fx1:1;
BYTE positionInCartesianCoordinates:1;
BYTE measuredPositionInPolarCoordinates:1;
BYTE positionInWGS84Coordinates:1;
BYTE timeOfDay:1;
BYTE targetReportDescriptor:1;
BYTE messageType:1;
BYTE dataSourceIdentifier:1;
```

```
}fspecData;
……
3.2各数据项结构定义
struct DataSourceIdentifier  //数据源识别
{
BYTE SIC;
BYTE SAC; //正常SAC始终为00
}dataSourceIdentifier;
struct TimeOfDay//当日时间
{
BYTE timeOfDay[3];
}timeOfDay;
struct PositionInWGS84Coordinates//WGS84坐标
{
BYTE longtitude[4];
BYTE latitude[4];
}positionInWGS84Coordinates;
……
```

由于ASTERIX数据格式的数据具备前后关联性,因此必须顺序读取,根据是否有字段判断是否需要读取。主要流程如下：

(1)读取前两个字段CAT和LEN;(2)根据FSPEC确定有哪些字段存在；(3)对每个存在的项目进行逐一循环,读取每个字段的数据。如果字段长度为固定长度,则直接读取；如果字段长度为变长,则根据数据结构格式确定最终的数据长度。(4)对每个字段进行处理,确定字段中每个元素的值；

由于完整解码过程比较复杂,请参考笔者合著的另外一篇论文《一种组装式ASTERIX数据解码算法》。

4 个性化数据定制输出

为了满足不同用户的信息需求,软件提供了个性化数据定制输出功能。用户可以根据自己的需求,选择需要输出的数据字段和格式。

选择数据字段：软件需提供数据字段选择界面,用户可以通过勾选或取消勾选来选择需要输出的数据字段。

设置数据格式：对于每个选中的数据字段,用户可以设置其输出格式,如数值型、字符串型等。此外,用户还可以设置数据的精度和单位等参数。

以及实现一些更个性化的数据定制,如通过查表的方式实现航班号三字码转二字码,实现代码(部分)如下。

```
int AirLineListPos =threeCodeList.indexOf(FightNum.left(3));
QString AirLine2;
if(AirLineListPos!=-1)
{
AirLine2=twoCodeList.at(AirLineListPos);
}
if(AirLine2=="")
{
in<<FightNum.trimmed()<<",";
}
else
{
in<<AirLine2<<FightNum.mid(3,4).trimmed()<<",";
}
```

5 可视化软件界面实现软件解码及数据输出的状态监控及参数配置

为了提供良好的用户体验和方便的操作,软件采用QT Creator编程软件和C++语言进行图形界面开发并实现以下功能。

主界面设计：软件主界面包括数据接收状态监控窗口、FTP上传状态监控窗口、软件状态日志显示窗口等。

参数配置：用户可以设置多点定位数据源的IP地址及端口号、输出文件的保存路径和名称等。

错误处理及状态监控：软件在解码和输出过程中会进行错误检测和处理。当遇到错误时,软件会在状态监控区显示错误信息并在状态日志显示窗口显示。

软件日志：记录软件在运行过程中的详细日志信息,包括启动时间、停止时间、操作记录、错误信息等。

6 总结

本文介绍了一款多点定位数据解码及格式转换软件的设计与实现。该软件采用C++语言和QT Creator编程软件进行开发,实现了UDP实时录取、数据解码、个性化数据定制输出和可视化软件界面等功能。通过该软件,用户单位可以更加便捷地获取所需的场面航班信息,实现信息的实时共享和高效利用。这不仅有助于提升机场的安全保障水平,还能优化机场的运营流程,提高服务质量和效率。

[参考文献]

[1]冯超.ASTERIX协议数据解析算法研究与实现[J].现代导航,2021,12(6):444-448.

[2]干浩亮,吴世桂,李运生,等.多点定位系统数据格式分析与研究[J].现代导航,2021,12(1):37-45.

[3]杨梦菲.多点定位系统及关键技术[J].中国科技信息,2022(10):40-41.

作者简介：

李建宏(1990--),男,汉族,重庆市渝北区人,大学本科,职称：工程师,研究方向：空中交通管理通信、导航、监视。