Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 5, No. 3, 2023

Receiving ADS-B Signals on Embedded Linux using RTL-SDR: A Practical Guide

Songyin Tan, Hongping Pu *

Sichuan University of Science & Engineering, CO 10622, China
* **Corresponding author**: Hongping Pu (Email: 348108700@qq.com)

Abstract: This paper presents a solution for ADS-B signal reception with RTL-SDR based on embedded Linux. This system utilizes an inexpensive RTL-SDR radio receiver for real-time reading of aeronautical messages, and it is experimentally demonstrated that this combination provides a significantly cost-effective way to track aircraft. By using an embedded Linux system, this paper demonstrates the high degree of portability and automation that can be achieved with RTL-SDR and its built-in ADS-B signal reception capability. The solution can be used in a wide range of applications and has high practical value.

Keywords: Embedde; RTL-SDR; Linux; ADS-B.

1. Introduction

Air traffic, as an essential part of modern society, has become increasingly busy with the rapid growth of global travel and cargo transportation. In order to ensure that aircraft can take off, land and fly safely, it is crucial to obtain timely and accurate information about the position and status of the aircraft. Traditional air traffic monitoring systems, such as radar, have been widely adopted but face limitations such as high cost, complex deployment and restricted coverage.[1] Therefore, it has become an urgent need to promote the development and application of a low-cost and efficient air traffic monitoring technology. ADS-B (Automatic Dependent Surveillance-Broadcast) technology, as an important innovation in modern aviation, is gradually replacing the traditional radar monitoring system. It realizes the ability to passively monitor the aircraft position, speed, altitude and other key data through the wireless signals transmitted by the aircraft itself.

Based on the embedded Linux platform and RTL-SDR (Software Defined Radio), we can build a low-power, cost-effective ADS-B monitoring system. Such a system not only realizes real-time reception and decoding of ADS-B signals, but also effectively provides complete monitoring of aircraft position and status information. As the embedded Linux platform is flexible and easy to develop and customize, while RTL-SDR has excellent reception performance and a wide range of application areas, the combination of the two can bring a revolutionary breakthrough in air traffic monitoring.

Traditional air traffic monitoring systems, such as those based on radar technology, provide reliable aircraft positioning and tracking, but face several limitations and challenges. First, the deployment of these systems requires expensive equipment and complex infrastructure, resulting in high costs. This limits the popularity and scope of application of the systems, especially in resource-constrained regions or countries. Second, traditional systems have limited coverage, especially in rural areas far from radar sites or over oceans, where monitoring is inaccurate and does not provide global aerial surveillance.[2]

To solve these problems, ADS-B technology has been widely studied and applied in recent years. However, there are still some challenges and limitations in the current air traffic monitoring system using ADS-B technology. On the one hand, although existing ADS-B ground stations are capable of receiving and decoding ADS-B signals, they usually require expensive and specialized hardware equipment, which limits their popularity and application scope.[3] On the other hand, in some areas or situations, traditional ground stations cannot provide sufficient coverage or real-time data updates, leading to inaccuracies and delays in monitoring.

In addition, the ADS-B monitoring system using embedded Linux platform in combination with RTL-SDR technology faces some challenges. First, it is a challenge to configure and optimize the RTL-SDR hardware device on the embedded platform to obtain stable and accurate ADS-B signals. Second, the reception sensitivity and performance of RTL-SDR may be affected by environmental interference, signal blocking, etc., which need to be optimized and improved to address these issues. Finally, how to design and implement efficient signal decoding algorithms and process large amounts of aircraft data to obtain reliable and real-time aerial surveillance results is also one of the key issues to be addressed.

In view of these challenges and limitations, this thesis aims to explore and provide an efficient method to receive ADS-B signals on an embedded Linux platform using RTL-SDR. By overcoming the challenges in hardware configuration, signal interference and data processing, we hope to solve the current problems faced by ADS-B monitoring systems and provide a practical solution for establishing a low-cost and high-performance air traffic monitoring system.[4]

2. Related Research

2.1. Overview of ADS-B Technology

2.1.1. Sub-section Headings

TheADS-B (Automatic Dependent Surveillance-Broadcast) is a wireless communication technology used to exchange data between drones and between aircraft and ground stations. The following is one way to describe ADS-B technology in detail: ADS-B technology is based on automatic dependent surveillance and broadcasting equipment installed on UAVs, which enables the aircraft to periodically broadcast its precise position, speed, heading, altitude, and other critical information. These broadcast messages can be transmitted to nearby aircraft and ground stations via various communication links, enabling collaboration and communication between them. ADS-B technology offers many advantages over traditional radar systems. First, ADS-B does not rely on ground facilities, and aircraft can directly broadcast messages to other aircraft and ground stations, allowing monitoring to extend from the visual range of a radar site to a global scale. This broadcast data exchange based on the station sending mode makes monitoring more reliable and allows accurate aircraft position information to be obtained even in areas without radar coverage.[5] Second, ADS-B provides more frequent and accurate aircraft data updates, typically broadcasting dozens of times per second. This allows air traffic managers to obtain real-time and accurate flight conditions to optimize traffic flow and provide more efficient flight planning. In addition, ADS-B technology offers the advantages of lower cost and ease of deployment. Relative to traditional ground-based radar systems, ADS-B employs broadcasting based on equipment on board aircraft, eliminating the need for expensive infrastructure, reducing construction and maintenance costs, and facilitating the popularization and use of the technology.[6] However, despite the importance of ADS-B technology in improving aeronautical surveillance, there are some limitations. One of the main challenges is to ensure the security and reliability of ADS-B messages to prevent data tampering and spoofing attacks. Current research is devoted to the development of encryption and authentication mechanisms to protect the integrity and trustworthiness of ADS-B messages.

In conclusion, ADS-B, as an advanced aviation surveillance technology, brings great potential to the field of aviation by broadcasting aircraft information in real time and expanding the surveillance range. It can improve air traffic management, traffic efficiency and safety while reducing costs and resource consumption.[7]

2.2. Embedded Linux Platform and RTL-SDR Technology

The embedded Linux platform is a solution for implementing operating system functionality on resource-constrained devices. It offers advantages such as flexibility, customizability, and low power consumption, making it ideal for building a variety of embedded systems. Meanwhile, RTL-SDR (Realtek Software Defined Radio) technology, a software-defined radio technology, is widely used to receive and process a wide range of wireless signals. The following is one way to describe the embedded Linux platform with RTL-SDR technology in detail.

2.2.1. Embedded Linux Platform:

An embedded Linux platform is a special configuration of the Linux-based operating system designed to accommodate the resource-constrained performance volume requirements of embedded devices.[8] Compared to traditional desktop or server systems, embedded Linux platforms can run on low-power devices such as microprocessors, system-on-chips, or single-board computers. Embedded Linux platforms offer the following key features and benefits:

Flexibility: Embedded Linux platforms can be customized and configured according to specific application needs to meet different hardware requirements and functional needs. Developers can precisely control the required components and functions according to project requirements, thus reducing system resource consumption and power consumption.

Customizability: The embedded Linux platform provides powerful customization capabilities that enable developers to select and adjust to specific device, target market or application requirements. This flexibility makes the embedded Linux platform ideal for adapting to a variety of different embedded devices and application scenarios.

Low Power Consumption: Embedded Linux platforms are optimized to work in resource-constrained environments and often employ energy-saving strategies to extend the battery life of devices. System power consumption can be minimized by optimizing the kernel, shutting down unnecessary services and processes, and using low-power hardware components and drivers.

2.2.2. RTL-SDR Technology:

The RTL-SDR (Realtek Software Defined Radio) is a software-defined radio technology based on real-time scanning devices, which utilizes a TV tuner chip (usually Realtek-RTL2832U) to transform a USB-interface computer into a powerful RF receiver. The main principle of the RTL-SDR technology is to input an antenna signal into the TV tuner chip and then use software to decode and process the received signal. The main principle of RTL-SDR technology is to input antenna signals into a TV tuner chip and then use software to decode and process the received signals. This allows the user to receive and analyze a wide range of radio signals, including FM broadcasts, aeronautical communications, radio signals, ADS-B signals, and more.

The use of RTL-SDR technology in aviation has been widely explored. Through the use of RTL-SDR devices and associated software, it is possible to receive and decode ADS-B signals to obtain the position, speed, and other information of an aircraft. This software-defined radio-based approach has the advantages of low cost and high flexibility over the traditional hardware circuit and device-based approach, and can provide sufficient performance to meet most aeronautical monitoring requirements.[9]

In summary, the combination of the embedded Linux platform and RTL-SDR technology provides a low-cost, customizable, and efficient solution for aeronautical monitoring. The embedded Linux platform provides robust operating system functionality and flexibility, while the RTL-SDR technology makes it ideal for receiving and processing a wide range of wireless signals, and is particularly suitable for receiving ADS-B signals and enabling air traffic monitoring.

3. Methodologies

3.1. Reliminary

SDR choose RTL2832U+R820T2, its frequency range is 24-1766Mhz, its acceptance ability is relatively strong in the market SDR. Linux development board choose I.MX6U mini development board, operating system choose Debian operating system.

The software is the dump1090 maintained by FlightAware, which is installed on the Fusion N1, which has two USB ports and strong CPU performance, and can theoretically start two SDRs.

Here, the open source WebSDR page is chosen to save some time in designing the Ui interface, the RTL-SDR number used for the WebSDR page is 0, and the RTL-SDR number used for ADS-B reception is 1. Regarding how to get the RTL-SDR device number, a relatively simple method is given here. First, make sure that the previous RTL-SDR is being used; insert a new RTL-SDR; execute the command rtl_test on the terminal, and not surprisingly, the following display is obtained, as you can see, rtl_test detects two RTL-SDRs and indicates that it cannot be opened because device number 0 is occupied. We can now conclude that device 1 is the newly inserted RTL-SDR.

```
root@BelovedZY:~# rtl_test
Found 2 device(s):
0: Realtek, RTL2838UHIDIR, SN: 00000001
1: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
usb_claim_interface error -6
Failed to open rtlsdr device #0.
```

Figure 1. Get RTL-SDR device number code

Make a note of the number 1 as the newly inserted RTL-SDR.

Finally, the dump1090 page uses Frp intranet penetration to port 8082 on the server, and then is reverse proxied via Nginx to port 80 on the server, hooked up to CloudFlare for human access.

3.2. Compile and Install

dump1090 relies on RTL-SDR drivers and support libraries to work, while RTL-SDR connects to the host via USB, so it is necessary to install three dependencies: rtl-sdr, librtlsdr-dev, libusb-1.0-0-dev. Since you need to pull the dump1090 source code from GitHub to compile and package it into a deb file, you also need to install git, build-essential, pkg-config, debhelper, dh-systemd, libncurses5-dev, libbladerf-dev. In addition, the dump1090 maintained by FlightAware exists as a module of Lighttpd, so lighttpd must be installed as well. To summarize, run the following command in the terminal to install the above software.

```
:~# apt update
:~# apt install build-essential debhelper rtl-sdr libusb-1.0-0-dev librtlsdr-dev pkg-config dh-systemd libncurses5-dev libbladerf-dev git lighttpd -y
```

Figure 2. Installation of lighttpd software code

Pull the dump1090 source code locally, then go to the source directory and compile and package it.

```
:~# cd /home
:/home# git clone https://github.com/flightaware/dump1090
:/home# cd dump1090
:/home/dump1090# dpkg-buildpackage -b
```

Figure 3. Pulling dump1090 source code

After a few moments, you will find the packaged deb file in the previous directory.

```
:/home/dump1090# cd ../
:/home# ls
dump1090                    dump1090-fa_3.8.0_arm64.buildinfo
dump1090_3.8.0_all.deb      dump1090-fa_3.8.0_arm64.changes
dump1090-fa-dbgsym_3.8.0_arm64.deb      dump1090-fa_3.8.0_arm64.deb
```

Figure 4. Deb file code

The filenames may vary depending on the CPU architecture, but they are all pretty much the same, so make a note of the filenames and install them separately.

```
:/home# vi /etc/default/dump1090-fa
```

Figure 5. dump1090 configuration file code

First of all, configure dump1090 to boot up, directly assign the ENABLED field in the configuration file to yes.

Since you want to specify dump1090 to use RTL-SDR No.1, you need to modify the RECEIVER_OPTIONS field, add --device-index 1 --gain 50 to tell dump1090 to turn on SDR No.1 with a gain of 50dB. gain depends on the actual situation; positive or negative value is acceptable.

The last thing is to tell dump1090 the latitude and longitude, you can fill in an approximate location for safety reasons. I am at latitude 30°383008″, longitude 103°394131″, so add --lat 30°383008 --lon 103°394131 to the DECODER_OPTIONS field.

To summarize the above requirements, a sample configuration file is given here.

```
# dump1090-fa configuration
# This is sourced by /usr/share/dump1090-fa/start-dump1090-fa as a
# shellscript fragment.

# If you are using a PiAware sdcard image, this config file is regenerated
# on boot based on the contents of piaware-config.txt; any changes made to this
# file will be lost.

# dump1090-fa won't automatically start unless ENABLED=yes
ENABLED=yes
RECEIVER_OPTIONS="--device-index 1 --gain 50 --ppm 0"
DECODER_OPTIONS="--lat 30.383008 --lon 103.394131 --max-range 360 --fix"
NET_OPTIONS="--net --net-heartbeat 60 --net-ro-size 1300 --net-ro-interval 0.2 --net-ri-port 0 --net-ro-port 30002 --net-sbs-port 30003 --net-bi-port 30004,30104 --net-bo-port 30005"
JSON_OPTIONS="--json-location-accuracy 1"
```

Figure 6. Configuration File Sample Code

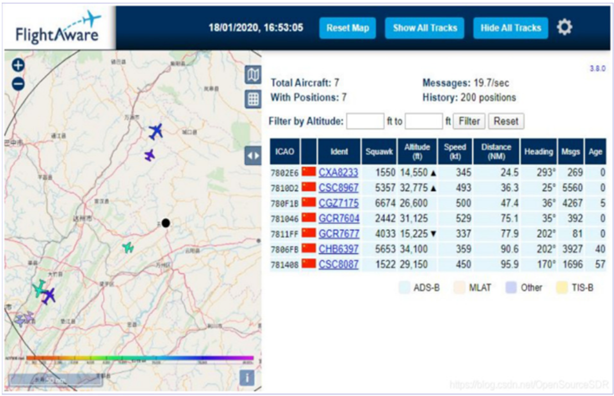

Figure 7. Home view

4. Conclusion

In this paper, a solution for ADS-B signal reception with RTL-SDR based on embedded Linux has been presented. This system utilizes an inexpensive RTL-SDR radio receiver for real-time reading of aeronautical messages, and experiments have demonstrated that this combination provides a significantly cost-effective way to track aircraft.

By using an embedded Linux system, this paper demonstrates the high degree of portability and automation that can be achieved with RTL-SDR and its built-in ADS-B signal reception capability. The solution can be used in a wide range of applications and is of high practical value.

However, note that this technology may be subject to some constraints and limitations in some cases. For example, the reception range of RTL-SDR may be affected by weather conditions and radio airspace interference. Future research could consider ways to ameliorate the effects of these limitations to improve system stability and performance.

Overall, this research provides a novel and effective way to enable ADS-B signal reception with inexpensive, easily configurable hardware devices and embedded Linux-based RTL-SDR. It opens new possibilities for aviation regulators, pilots and hobbyists alike by providing a cost-effective method for real-time flight tracking.

Then, we expect to further improve the adaptability, flexibility and scalability of the system in future exploration and research to promote the development of civilian aviation monitoring technology based on low-cost hardware.

References

[1] Lv Jingjian,Xiao Haiqiao. An overview of embedded systems for the 21st century[J] . Electronic Quality,2001(08).

[2] Zhao Tianlong. Development and Application of Embedded Linux System[J] . Western Leather. 2017,39(06).

[3] Zheng Hong. Research on application program design based on embedded system[M]. Xi'an University of Electronic Science and Technology,2007.

[4] Ningbo.Research and realization of Linux embedded system in industrial control field[M]. University of Electronic Science and Technology, 2007.

[5] Xiong Jianfang, Gao Ji, Ren Heyu. Application of Embedded System in Intelligent Transportation[J]. Modern Electronic Technology,2006(13):129-131.

[6] YIN Hu, YU Juan, QI Mingzhu. Application of embedded system in smart home environment[J]. Electronic Technology and Software Engineering,2022(15),29-32.

[7] TANG Xiaoying,LIU Zhiwen,LIU Weifeng,MENG Xu. Embedded system and its application in medical instrumentation [J]. Electronic Technology Applications,2008 (04):7-9+15.

[8] Li Houliangzhe. Research on the application of embedded real-time operating system in small unmanned aerial vehicle flight control system[J].Proceedings of the 2014(5th) China UAV Conference,2014.

[9] YAN Ying. Development status and prospect of embedded system[J]. Science Chinese,2016(09):188-190.