Konzept ADS-B Viewer

April 2, 2020

Designentscheidungen

ADS-B Empfang

Hardware

- RTL-SDR Dongle
- Rasberry-Pi

Software

- Kommandozeile Rasberry-Pi: while true; do ((echo HTTP/1.0 200 ok; echo; rtl_adsb -S -Q2;) | netcat -l 1234); done
- Änderungen:
  - Erkennung der ADS-B Meldungen verbessern (pufferübergreifende Auswertung)
  - Fehlerkorrekturen von [2] übernehmen

Visualisierung

Hardware

- ThinkPad X1 Laptop Windows 10

Software

- C++
- MFC mit Document/View-Konzept benutzen. Immer nur ein Dokument. Zunächst mit nur eine Ansicht auf die Rawmessages und später eine Ansicht der dekodierten ADS-B Meldungen.

  The view is responsible for displaying and modifying[1] the document's data but not for storing it.[2] The document provides the view with the necessary details[3] about its data. You can let the view access the document's data members directly, or you can provide member functions in the document class for the view class to call.

  When a document's data changes, the view responsible for the changes typically calls the CDocument::UpdateAllViews function for the document, which notifies all the other views by calling the OnUpdate member function for each. The default implementation of OnUpdate invalidates the view's entire client area. You can override it to invalidate only those regions of the client area that map to the modified portions of the document.

- LibCURL für den Netzwerkzugriff benutzen

[1] gemeint ist: vom User!
[2] gemeint ist: in eine Datei speichern!
[3] also z.B. die Dekodierung im Dokument machen!

- Designdetails:
  - Das Document erzeugt einen Worker-Thread (in der Klasse ADSBThread), der den ADS-B Rohdatenstrom von der Netzwerkschnittstelle liest.
  - Der Rohdatenstrom kommt in Blöcken variabler Länge an. Die enthaltenden Informationen erstrecken sich evtl. über die Blockgrenzen hinweg. Dies muss beim Parsen (in der Klasse CombineChunks) berücksichtigt werden um keine Informationen zu verlieren.
  - Nach jedem neu ankommenden Chunk des Rohdatenstromes werden die darin enthaltenen Rawmessages geparst, d.h. in Zeitmarken und Rawmessages zerlegt, und in einer Queue des Documents gespeichert. Alle empfangenen ADS-B Messages sind in dieser Queue persistent.
  - Die Klasse ADSBThread erbt von der Parser-Klasse CombineChunks.
  - Die neu ankommenden ADS-B Messages werden mit der Message ADSB_MESSAGE_ID zum View-Window gesendet, dort mit OnUser() empfangen, und visualisiert.
  - Dazu wird die View von CListView abgeleitet und mittels der AddItem-Methode die neuen Queueelemente hinzugefügt.
  - Um Speicherlecks beim Beenden der Applikation zu vermeiden, wird dem Thread im Destructor des Documents signalisiert, dass er sich beenden soll (durch setzten von loopBreak=true) dann wird auf das Event threadFinished gewartet.
  - Dekodierung der ADS-B Rawmessages: siehe [2]
- Softwaretest: Die pufferübergreifende Auswertung der Informationsblöcke ist komplex. Zum Test habe ich ein Tool geschrieben: ADSBTest. Dieses Tool könnte auch für die Empfangsseite genutzt werden.

Analyse rtl-adsb

- Die Samplezeit ist 0.5 μs, die Sampleauflösung ist 8 Bit.
- Die Sampledaten vom Demodulator kommen in einem Byte-Puffer der Länge 16 * 16384 = 262144 an.
- Jeder Samplezeitpunkt wird durch 2 Bytes beschrieben: i und q.
- 1. Schritt: die beiden Bytes in ein Wort wandeln: $i^2 + q^2$. Die Länge des Puffers reduziert sich dadurch auf 131072 Worte.
- Funktion manchester():
  - Die Wörter der Puffers werden nacheinander daraufhin untersucht, ob sie eine $8\mu s$, d.h. 16 Worte lange Präambel [2] enthalten.
  - Die Präambel ist durch folgende Bit-Abfolge definiert: 1010000101000000, d.h. die Bits 0,2,7 und 9 müssen 1 sein, der Rest 0.
  - Diese Prüfung wird durch die Funktion preamble() vorgenommen. Eine 1 wird returnt, falls eine Präambel gefunden wurde, sonst 0.
  - Zum Testen der Funktionalität habe ich 2 Dateien generiert: Rawmessages in Textform "ADSB-Messages.txt" und die dazugehörigen binären Rohsamples "ADSB-Dump.bin". Die beiden Dateien befinden sich im Verzeichnis Projekte/rtl-sdr.

ADS-B Datenformatbeschreibung

Es gibt die folgenden Downlink Formate:

DF	Beschreibung
0	Airborne Collision Avoidance System (ACAS), Traffic Alert and Collision Avoidance System (TCAS)
1-3	Reserviert
4	Antworten auf Abfragen von bodengestützten Sekundärradaranlagen, Surveillance altitude
5	Antworten auf Abfragen von bodengestützten Sekundärradaranlagen, Surveillance ident
6-10	Reserviert
11	All-Call Reply (Acq. Squitter if II=0)
12-15	Reserviert
16	Long air-air surveillance (ACAS)
17	Extended Squitter
18	Extended Squitter, supplementary
19	Militärischer Extended Squitter
20	Comm-B, Mode-S EHS, Altitude
21	Comm-B, Mode-S EHS, Ident
22	militärisch genutzt
23	Reserviert
24	Comm-D Extended Length Message (ELM), Format siehe [3]

Parity Coding

Für ADS-B wird ein pseudo-zyklischer Code zur Bestimmung der Parität benutzt.
Das Generatorpolynom dieses Codes ist durch

$$G(x) = x^{24} + x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{10} + x^3 + x$$

gegeben [5], es gilt $x \in \{0, 1\}$.

Polynom-Multiplikation/Division

Es werden die Zusammenhänge aus [6] zusammengefasst. Leere Matrixelemente sind als als 0 zu interpretieren.
Die Menge der Polynome über dem Körper K und des Grades n sei gegeben durch $P_n = \{p_0 + p_1 x + \cdots + p_n x^n, x \in K\}$.
Die Multiplikation eines der Elemente $p(x) \in P_n$ mit einem Polynom $a(x) = a_0 + a_1 x + \cdots + a_m x^m$ des Grades m definiert eine lineare Abbildung M_{pa} von $P_n(x)$ in $P_{n+m}(x)$. Die Koeffizienten $\underline{b}_{pa} = (b_0, \cdots, b_{n+m})^T$ des Produkt-Polynoms

$$(pa)(x) = (ap)(x) = b_0 + b_1 x + \cdots b_{n+m} x^{n+m}$$

ergeben sich zu

$$\underline{b}_{pa} = M_{pa} \underline{p}$$

mit dem Vektor $\underline{p} = (p_0, \cdots, p_n)^T$ und der $((n+m+1) \times (n+1))$-Abbildungsmatrix (Multiplikationsmatrix)

$$M_{pa} = \begin{pmatrix} a_0 & & & & \\ \vdots & a_0 & & & \\ a_{m-1} & \vdots & a_0 & & \\ a_m & a_{m-1} & \vdots & \ddots & \\ & a_m & a_{m-1} & & a_0 \\ & & a_m & \ddots & \vdots \\ & & & \ddots & a_{m-1} \\ & & & & a_m \end{pmatrix}.$$

Mit der $((n+m+1)\times m)$-Matrix

$$A = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ & & \end{pmatrix}$$

kann man die für $a_m \neq 0$ nichtsinguläre $((n+m+1)\times(n+m+1))$-Matrix (Divisionsmatrix)

$$D = \begin{pmatrix} A, & M_{pa} \end{pmatrix}$$

definieren. Es gilt mit dem Vektor $\underline{r} = (r_0, \cdots, r_{m-1})^T$

$$D \begin{pmatrix} \underline{r} \\ \underline{a} \end{pmatrix} = \underline{b}_{pa}$$

Um diese mathematischen Erkenntnisse nutzen zu können, muss der Menge $\{0,1\}$ eine Körperstruktur zugeordnet werden können. Im Anhang wird bewiesen, dass dies möglich ist, wenn man mit der Körperaddition $+$ die Exclusiv-Oder-Operation $\oplus$ identifiziert und mit der Körpermultiplikation $\circ$ die Und-Operation $\wedge$. Weiterhin wird für diesen Körper bewiesen, dass seine inversen Matrizen A^{-1} mit verschwindender Spur $\mathrm{spur}\,(A)$ selbstinvers sind, dass also gilt $A^{-1} = A$. Die Spur der oben definierten Divisions-Matrizen verschwindet in diesem Körper, so dass eine evtl. rechenaufwendige Matrizeninversion entfällt.

DF17/DF18

DF20/DF21

Diese beiden Downlink-Formate sind Antworten, die von der Bodenstation aktiv abgefragt werden. Welche Daten übertragen werden hängen von einem BDS (Binary Data Selector) ab, der aber nicht in der Antwort enthalten ist.

Es gibt die folgenden BDS-Codecs:

BDS 2,0	Aircraft identification
BDS 2,1	Aircraft and airline registration markings
BDS 4,0	Selected vertical intention
BDS 4,4	Meteorological routine air report
BDS 5,0	Track and turn report
BDS 6,0	Heading and speed report

Die DF20/21 Messages sind wie folgt strukturiert:

Bezeichnung	Bits	Bedeutung
DF	5	Downlink Format
FS	3	Flight Status
DR	5	Downlink Request
UM	6	Utility Message
AC/ID	13	Altitude Code (DF20) or Identity (DF21)
MB	56	Comm-B Message
AP/DP	24	Address Parity/Data Parity

Aircraft identification BDS 2,0

Das Comm-B Messagefeld MB hat die folgende Struktur:

Inhalt	0x20	C8	C7	C6	C5	C4	C3	C2	C1
Bits	8	6	6	6	6	6	6	6	6

Die jeweils 6 Bits $b_n, \quad n = 0, \cdots, 5$ eines Zeichens $C_m = (b_5, \cdots, b_0), \quad m = 1, \cdots, 8$ werden als 0-basierter Index $\sum_{n=0}^{5}(2^n b_n)$ in die folgende Lookup-Tabelle interpretiert:

"#ABCDEFGHIJKLMNOPQRSTUVWXYZ#####␣###############0123456789######".

Das #-Zeichen bezeichnet einen ungültigen Index, das ␣-Zeichen ist das Leerzeichen.
Man erhält die Aircraft Identification die Zeichenkette C8,C7,C6,C5,C4,C3,C2,C1.

Anhang

Körpereigenschaften

Eine Menge Π bildet einen Körper, wenn auf ihr die zweistelligen Operationen der Addition $+$ und der Multiplikation $\circ$ definiert sind und die drei Eigenschaften

1. $\left(\Pi, \quad + \right)$ muss eine kommutative Gruppe mit dem neutralen Element $e = 0$ sein, d.h. es muss gelten
   - Assoziativgesetz: $\forall a, b, c \in \Pi$ gilt $a + (b + c) = (a + b) + c$.
   - Kommutativgesetz: $\forall a, b \in \Pi$ gilt $a + b = b + a$.
   - Neutrales Element e: $\forall a \in \Pi$ gilt $a + e = a$.
   - Inverses Element: $\forall a \in \Pi$ gibt es ein inverses Element a^{-1} mit $a + a^{-1} = e$.
2. $\left(\Pi \setminus \{0\}, \quad \circ \right)$ muss eine kommutative Gruppe mit dem neutralen Element $e = 1$ sein, d.h. es muss gelten
   - Assoziativgesetz: $\forall a, b, c \in \Pi$ gilt $a \circ (b \circ c) = (a \circ b) \circ c$.
   - Kommutativgesetz: $\forall a, b \in \Pi$ gilt $a \circ b = b \circ a$.
   - Neutrales Element e: $\forall a \in \Pi \setminus \{0\}$ gilt $a \circ e = a$.
   - Inverses Element: $\forall a \in \Pi \setminus \{0\}$ gibt es ein inverses Element a^{-1} mit $a \circ a^{-1} = e$.
3. Distributivgesetze: $\forall a, b, c \in \Pi$ gilt $a \circ (b + c) = a \circ b + a \circ c$ und $(a + b) \circ c = a \circ c + b \circ c$

erfüllt sind.

Behauptungen

1. Identifiziert man mit der Addition $+$ die Exklusiv-Oder-Operation $\oplus$, so ist $\left(\Pi, \quad + \right)$ eine kommutative Gruppe mit dem neutralen Element 0.
2. Identifiziert man mit der Multiplikation $\circ$ die Und-Operation $\wedge$, so ist $\left(\Pi \setminus \{0\}, \quad \circ \right)$ eine kommutative Gruppe mit dem neutralen Element 1.
3. Ausserdem gelten die beiden Distributivgesetze für die Körperelemente.

Beweise

Es gibt maximal drei Variablen, sodass es mir am einfachsten und verständlichsten erscheint, die folgenden Beweise auf Grundlage von Wahrheitstafeln zu führen.

Beweis der Behauptung 1

Mit der Identifizierung der Addition $+$ mit der Exklusiv-Oder-Operation $\oplus$ erhält man für das Assoziativgesetz $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ und für das Kommutativgesetz $a \oplus b = b \oplus a$.
Mit der Wahrheitstabelle der Exklusiv-Oder-Operation 1

$a \oplus b$	a	b
0	0	0
1	1	0
1	0	1
0	1	1

Table 1: Wahrheitstabelle Exclusiv-Oder (XOR)

und den beiden Tabellen 2 und 3 sieht man, dass das Assoziativgesetz erfüllt ist.

$a \oplus (b \oplus c)$	$b \oplus c$	a	b	c
0	0	0	0	0
1	0	1	0	0
1	1	0	1	0
0	1	1	1	0
1	1	0	0	1
0	1	1	0	1
0	0	0	1	1
1	0	1	1	1

Table 2: zum Assoziativgesetz der Exklusiv-Oder-Operation (XOR)

$(a \oplus b) \oplus c$	$a \oplus b$	a	b	c
0	0	0	0	0
1	1	1	0	0
1	1	0	1	0
0	0	1	1	0
1	0	0	0	1
0	1	1	0	1
0	1	0	1	1
1	0	1	1	1

Table 3: zum Assoziativgesetz der Exklusiv-Oder-Operation (XOR)

Das Kommutativgesetz ist erfüllt, wie man der Wahrheitstabelle 1 direkt entnehmen kann.

Das neutrale Element 0 erfüllt die Beziehung $a \oplus 0 = a$ (siehe 1).

Das inverse Element zu 0 ist 0, da $0 \oplus 0 = 0$ gilt, das inverse Element zu 1 ist 1, da $1 \oplus 1 = 0$ gilt.

Damit sind alle notwendigen Eigenschaften der Addition nachgewiesen und $\left(\ \Pi, \ \ + \ \right)$ ist eine kommutative Gruppe mit dem neutralen Element 0.

Hinweis: Die beiden Zahlen 0 und 1 sind selbstinvers bezogen auf die Körperaddition. Aus $c = a \oplus b$ folgt $a = c \oplus b$ also ist auch die Addition $\oplus$ selbtsinvers.

Beweis der Behauptung 2

Mit der Identifizierung der Multiplikation $\circ$ mit der Und-Operation $\wedge$ erhält man für das Assoziativgesetz $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ und für das Kommutativgesetz $a \wedge b = b \wedge a$. Beide Gesetze sind erfüllt (siehe [7]).

Für das neutrale Element 1 gilt $1 \circ 1 = 1$, wie man der Wahrheitstabelle 4 entnehmen kann.

$a \wedge b$	a	b
0	0	0
0	1	0
0	0	1
1	1	1

Table 4: Wahrheitstabelle der Und-Operation (AND)

$\overline{a \wedge b}$	a	b
1	0	0
1	1	0
1	0	1
0	1	1

Table 5: Wahrheitstabelle Negation Und-Operation (NAND)

Das inverse Element zu 1 ist 1, da $1 \wedge 1 = 1$ gilt.

Damit sind alle notwendigen Eigenschaften der Multiplikation nachgewiesen und $\left(\ \Pi, \ \circ \ \right)$ ist eine kommutative Gruppe mit dem neutralen Element 1.

Hinweis: Die Zahl 1 ist selbstinvers bezogen auf die Körpermultiplikation.

Beweis der Behauptung 3

Für die beiden Distributivgesetze für die Körperelemente erhält man mit den Operationen $\oplus$ und $\wedge$ die beiden Bedingungen

$$a \wedge (b \oplus c) = a \wedge b \oplus a \wedge c$$

und

$$(a \oplus b) \wedge c = a \wedge c \oplus b \wedge c.$$

Die beiden Distributivgesetze sind erfüllt, wie man den untenstehenden Wahrheitstabellen entnehmen kann.

$a \wedge (b \oplus c)$	$b \oplus c$	a	b	c
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
1	1	1	1	0
0	1	0	0	1
1	1	1	0	1
0	0	0	1	1
0	0	1	1	1

Table 6: zur Behauptung 3, 1. Distributivgesetz

$a \wedge b \oplus a \wedge c$	$a \wedge b$	$a \wedge c$	a	b	c
0	0	0	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
1	1	0	1	1	0
0	0	0	0	0	1
1	0	1	1	0	1
0	0	0	0	1	1
0	1	1	1	1	1

Table 7: zur Behauptung 3, 1. Distributivgesetz

$(a \oplus b) \wedge c$	$a \oplus b$	a	b	c
0	0	0	0	0
0	1	1	0	0
0	1	0	1	0
0	0	1	1	0
0	0	0	0	1
1	1	1	0	1
1	1	0	1	1
0	0	1	1	1

Table 8: zur Behauptung 3, 2. Distributivgesetz

$a \wedge c \oplus b \wedge c$	$a \wedge c$	$b \wedge c$	a	b	c
0	0	0	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	1	1	0
0	0	0	0	0	1
1	1	0	1	0	1
1	0	1	0	1	1
0	1	1	1	1	1

Table 9: zur Behauptung 3, 2. Distributivgesetz

Definition

Die Menge Π mit der Addition $\oplus$ und der Multiplikation $\wedge$ wird als Körper der binären Zahlen $\mathbb{B}\left(\ \Pi, \quad +, \quad \circ\ \right)$ bezeichnet. Der Körper $\mathbb{B}$ hat die Charakteristik 1.

Behauptung

Invertierbare Matrizen über dem Körper $\mathbb{B}\left(\ \Pi, \quad +, \quad \circ\ \right)$ der binären Zahlen sind selbstinvers, falls ihre Spur verschwindet.

Beweis

Muss noch erbracht werden.

References

[1] https://osmocom.org/projects/rtl-sdr/wiki

[2] https://github.com/watson/libmodes

[3] https://www.radartutorial.eu/13.ssr/sr24.de.html

[4] https://github.com/junzis/pyModeS/blob/master/pyModeS/decoder/common.py

[5] Fundamentals of Mode S Parity Coding; J. L. Gertz; Project Report ATC-117

[6] Polynomial GCDs by Linear Algebra; Barry Dayton; Northeastern Illinois University; March 2004

[7] https://de.wikipedia.org/wiki/Boolesche_Algebra