An open source and flexible ACARS receiver based on software defined radio

Baptiste Chamaillard ¹ Maxime Lastera ¹
Supervisor : Damien Roque ²

¹ISAE Supaero, Mastère TERA
²ISAE Supaero, Département Electronique, Optronique et Signal (DEOS)

4 mai 2017
The goal of this demonstration is to show some of the benefits offered by a **software-defined radio (SDR)** receiver:

- **simplicity** of implementation due to the "on-the-shelf" building blocks.
- **low material cost** and integrated hardware.
- **versatility and extensibility** of the receiver.

→ **excellent educational tool** to understand the receiver architecture.

For purposes of illustration, the discussion is based on a **concrete example of an ACARS receiver** developed during our final year project at ISAE-SUPAERO.
Project purpose:

- Theoretical model of a VHF ACARS transmission system (PHY & MAC layers)
- Implementation of a SDR receiver enabling real time decoding operations and providing human-readable informations.
Summary

1. Brief history and quick introduction to ACARS
   - A brief history
   - ACARS frame description

2. A short theoretical analysis of an ACARS transceiver
   - ACARS transmitter
   - ACARS receiver

3. Implementation of an ACARS receiver with GNU Radio
   - GNU Radio
   - General architecture
   - ACARSDecoder Block
   - ACARSParser Block

4. Demo
Summary

1. Brief history and quick introduction to ACARS
   - A brief history
   - ACARS frame description

2. A short theoretical analysis of an ACARS transceiver
   - ACARS transmitter
   - ACARS receiver

3. Implementation of an ACARS receiver with GNU Radio
   - GNU Radio
   - General architecture
   - ACARSDecoder Block
   - ACARSParser Block

4. Demo
A brief history

ACARS stands for Aircraft Communication Addressing and Reporting System.

- 1976: Use of VHF in North America at the unique frequency of 131.550 MHz.
- 1984: AIRCOM network which is an ACARS compatible network has been created by SITA.
- 1990: Use of Satellite link.

ACARS network overview
Use of ACARS network

<table>
<thead>
<tr>
<th>Flight Phase</th>
<th>From Aircraft</th>
<th>To Aircraft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Park/Taxi</td>
<td>LINK TEST</td>
<td>FLIGHT PLAN</td>
</tr>
<tr>
<td>Take-Off</td>
<td>ENGINE DATA</td>
<td>FLIGHT PLAN</td>
</tr>
<tr>
<td>Depart/ Climb</td>
<td>ENGINE DATA</td>
<td>WEATHER REPORT</td>
</tr>
<tr>
<td>En Route</td>
<td>DELAY INFO ETA</td>
<td>GATE ASSIGNMENT</td>
</tr>
<tr>
<td>Approach</td>
<td>GATE REQUEST</td>
<td>CONNECTING GATE</td>
</tr>
<tr>
<td>Land</td>
<td>ETA</td>
<td></td>
</tr>
<tr>
<td>Park/Taxi</td>
<td>FAULT DATA</td>
<td></td>
</tr>
</tbody>
</table>

ACARS communication according to the different flight phases
## ACARS frame format

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Nb of Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-key</td>
<td>16</td>
<td>Identification and synchronisation parameters</td>
</tr>
<tr>
<td>Bit sync</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Char sync</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>SOH</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Ack/Nak</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td>2</td>
<td>TEXT</td>
</tr>
<tr>
<td>Block ID</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>STX</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TEXT</td>
<td>220</td>
<td></td>
</tr>
<tr>
<td>ETX</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>BCS</td>
<td>2</td>
<td>End frame identification</td>
</tr>
<tr>
<td>BCS Suffix</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Summary

1 Brief history and quick introduction to ACARS
   - A brief history
   - ACARS frame description

2 A short theoretical analysis of an ACARS transceiver
   - ACARS transmitter
   - ACARS receiver

3 Implementation of an ACARS receiver with GNU Radio
   - GNU Radio
   - General architecture
   - ACARSDecoder Block
   - ACARSPARSER Block

4 Demo
ACARS transmitter

Overall architecture of the ACARS transmitter

1. A binary source $d[n]$.
2. A MSK (Continuous-Phase Modulation) modulator to convert the digital sequence $d[n]$ into an analogical signal $m(t)$ with $T_{\text{symbol}} = 1/1200$ s.
3. A DSB (Double Sideband) amplitude modulator to center the output signal on the carrier frequency $f_0 = 131,725$ MHz.
ACARS receiver

Overall architecture of the ACARS receiver

1. Oversampling & computing of the analogical input signal by the SDR receiver to derive its numerical complex envelope.
2. Non coherent amplitude demodulation (no need to recover the carrier and the phase).
3. Frame & symbol synchronization thanks to the frame preamble (pre-key).
4. MSK demodulation to detect the transmitted bits.
Summary

1 Brief history and quick introduction to ACARS
   - A brief history
   - ACARS frame description

2 A short theoretical analysis of an ACARS transceiver
   - ACARS transmitter
   - ACARS receiver

3 Implementation of an ACARS receiver with GNU Radio
   - GNU Radio
   - General architecture
   - ACARSDecoder Block
   - ACARSParser Block

4 Demo
GNU Radio

What is GNU Radio?

- **Free** and **open-source** software.
- **SDR applications** development tool.
- Pre-compiled signal processing blocks in **C** or **python**.

Why GNU Radio?

to offer to the community a flexible open-source ACARS receiver deployable with only **a VHF antenna** and a **SDR receiver**.
GNU radio receiver architecture

GNU radio receiver architecture

B. Chamaillard and M. Lastera

SDR ACARS Receiver

4 mai 2017 14 / 23
**Role** : Demodulate the received symbols in order to retrieve the transmitted bits when an incoming frame is detected.

**Parameters** : $samp\_rate$, $corr\_thres$

**Some programming features** :
- Involved in the frame detection and symbol synchronization steps.
- Block-processing of the data flow.
- Frame structure check-up and parity bits control $\rightarrow$ incremental improvement of the receiver robustness.
ACARSPARSER Block

**Role**: Make the decoded data understandable. Frame information is transmitted in binary format from **ACARSDecoder** block via **asynchronous** messages.

**Parameters**: `db_request_mode`, `API_Key`

**Processed content**:  
- Transmission mode.  
- Label.  
- Aircraft Address (HTTP request).  
- Aircraft flight number (IATA request).  
- Text.
Example of interpreted frame

New Frame!
Time: yyyy-mm-dd hh:mm:ss
Transmission Mode: E ==> Category B - Air to Ground transmission
Departure Airport: Heathrow (LHR) - Arrival Airport: Blagnac (TLS)
ACK/NAK: NAK
Label: 1 0
Block ID: 2
STX
Message Sequence Number: M 1 8 A
Flight Number: B A 0 3 7 9
Text: D R C 0 1 1 6 0 4 -- -- 1 5 4 6
End of Frame

New Frame!
Time: yyyy-mm-dd hh:mm:ss
Transmission Mode: e ==> Category B - Ground to Air transmission
Departure Airport: Heathrow (LHR) - Arrival Airport: Blagnac (TLS)
ACK/NAK: 2
Label: _ DEL ==> General response, demand mode, no information to transmit
Block ID: V
End of Frame

1. Date, Airplane Address and Airline company name have been changed to respect communication privacy
Example of interpreted frame

New Frame!
Time: yyyy-mm-dd hh:mm:ss
Transmission Mode: 2 ==> Category A - Broadcasting to all the stations
Departure Airport: Orly (ORY) - Arrival Airport: Houari Boumediene (ALG)
ACK/NAK: NAK
Label: H 1 ==> Message to/from terminal
Block ID: 5
STX
Message Sequence Number: C 0 2 A
Flight Number: Z I 0 2 5 9
Sublabel: # C F ==> Central Fault Display
Text: B W R N / W N 1 7 0 3 1 5 1 1 0 6 0 0 4 6 2 0 0 0 0 6
   D A T A L I N K   A T C   F A U L T
End of Frame

New Frame!
Time: yyyy-mm-dd hh:mm:ss
Transmission Mode: 2 ==> Category A - Broadcasting to all the stations
Departure Airport: Orly (ORY) - Arrival Airport: Houari Boumediene (ALG)
ACK/NAK: 5
Label: _ DEL ==> General response, demand mode, no information to transmit
Block ID: I
End of Frame

2. Date, Airplane Address and Airline company name have been changed to respect communication privacy

B. Chamaillard and M. Lastera
Example of interpreted frame

New Frame !
Time : yyyy-mm-dd hh:mm:ss
Transmission Mode : e ==> Category B - Transmitting from a specific station - Ground to Air link
Airplane Address : NUL - Aircraft Model : No Information Available (from https://planefinder.net)
Departure & Arrival Airports : No Information Available (from https://planefinder.net)
ACK/NAK : NAK
Label : S Q ==> Squitter Message
Block ID : NUL
STX
Version Number : 0 2
Service Provider : X S
IATA Station ID : T L S
ICAO Station ID : L F B O
Station Number ID : 0
Latitude : 4 3 3 7 N
Longitude : 0 0 1 2 3 E
Text : V 1 3 6 9 7 5 /
ETX

-----------------------------------------------
1. Brief history and quick introduction to ACARS
   - A brief history
   - ACARS frame description

2. A short theoretical analysis of an ACARS transceiver
   - ACARS transmitter
   - ACARS receiver

3. Implementation of an ACARS receiver with GNU Radio
   - GNU Radio
   - General architecture
   - ACARSDecoder Block
   - ACARSPARSER Block

4. Demo
Results:

- GNU Radio working receiver.
- Open-source project available at https://sourceforge.isae.fr/git/supacars

Enhancement and perspectives:

- Code optimization to reduce the receiver complexity.
- Study in depth ARINC 618 [1] and 620 [2] to enhance the receiver parsing features.
- Submit an article to the IEEE AESS Systems magazine.
Thank you for your attention. Do you have some questions?