Istanbul Medipol University
School of Engineering and Natural Sciences

Communication Systems – Spring 2025
Project Final Report

Hakan SABUNİŞ - Yusuf ÜNLÜ
DigiGuys

Project Leader: Hakan SABUNİŞ

Introduction

The main aim of this project was to use open-source software tools and inexpensive Software-Defined Radio (SDR) hardware to develop and implement a broadcast system that complies with the Digital Audio Broadcasting (DAB+) standard. Field tests using a commercial in-car DAB+ receiver successfully confirmed the established system's ability to broadcast digital audio over a single service. The theoretical benefits of DAB+ over conventional analog broadcasting are actually demonstrated by this implementation, which offers a real-world application of ideas like spectrum efficiency and robustness against noise, which are the cornerstones of contemporary digital communication systems.

Using the ODR-AudioEnc software, an audio source is first converted into the HE-AACv2 format as part of the operational flow of the previously stated broadcast chain. To create a transport stream that complies with standards, the compressed data stream is multiplexed using the ODR-DabMux component in a DAB ensemble structure (ETSI EN 300 401). Lastly, this stream is sent to the radio frequency (RF) layer via the ADALM-PLUTO SDR hardware after being modulated by the ODR-DabMod modulator using the Orthogonal Frequency Division Multiplexing (OFDM) approach. Both Yusuf Ünlü and Hakan Sabuniş worked together on the project's planning, integration, and reporting procedures.

The fundamental theoretical ideas covered in the "Communication Systems" course curriculum are synthesized in this complete project. Here, the practical implementation of Source Coding is the audio compression process; Multiplexing Techniques is the combination of data into a single packet; and Digital Modulation and Channel Coding practices are represented by OFDM based signal generation, convolutional coding, and time-interleaving operations. The SDR hardware's ability to transmit the system's output as an RF signal is a concrete example of how the concepts of transceiver systems and physical layer design are applied.

Blocks and Their Functionality

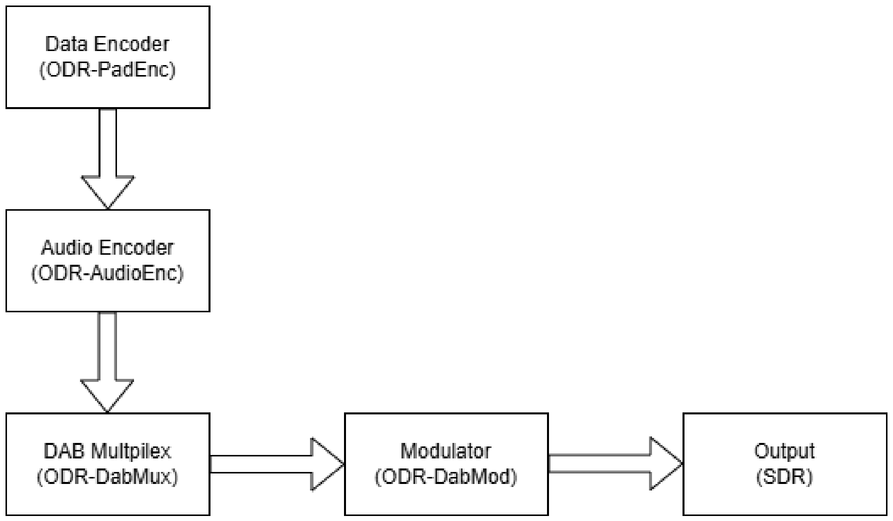

Data Encoder (ODR-PadEnc)
Audio Encoder (ODR-AudioEnc)
DAB Multpilex (ODR-DabMux)
Modulator (ODR-DabMod)
Output (SDR)

Figure 1: Block diagram of system.

Blocks and Their Functionality

As seen in Figure 1, the implemented broadcast system is a modular chain made up of three primary software blocks from the Open Digital Radio (ODR) toolset. These blocks process the signal in turn before the ADALM-PLUTO SDR hardware transmits it.

1. ODR-AudioEnc (Audio Encoder)
As the initial link in the broadcast chain, this block exemplifies the concepts of effectively encoding and digitally recording an audio signal. This procedure combines the concepts of source coding, quantization, and sampling.

A typical digital audio file is used to start the procedure, which is founded on the following ideas:
• Sampling: The original continuous audio wave was sampled 48,000 times per second for the input.wav file used in the project. This digital data is processed by the odr-audioenc tool at the designated 48000 Hz.

• Quantization: Using a certain bit depth (e.g., 16-bit), each sample was transformed into a numerical value. The raw PCM (Pulse-Code Modulation) data stream produced by this process has a very high bitrate that is inefficient for broadcasting, even though each audio sample is represented with great quality.

Broadcasting raw PCM data is inefficient. Source coding is useful in this situation. The PCM data is compressed by ODR-AudioEnc and saved in the DAB+ standard HE-AACv2 format. Perceptual audio coding, which eliminates audio components, is the method used by this codec. This enables the transmission of stereo music with good perceptual quality, even at low bitrates as 120 kbit/s.

2. ODR-DabMux (DAB Multiplexer)
One or more encoded audio streams are combined by the multiplexer into a single structured packet called a DAB "ensemble." Although Time-Division Multiplexing (TDM) is the foundation of this, the underlying structure is quite precise and mathematical. Mapping the different data services onto the DAB transmission frame structure is the main responsibility of the multiplexer.

Allocation of Capacity and Theoretical Framework Structure
Common Interleaved Frames (CIFs), the basic building blocks of the Main Service Channel (MSC), make up the continuous sequence into which the DAB multiplex is arranged. A CIF in DAB Mode I has a predetermined size of 55,296 bits and a defined duration of 24 ms.

This frame's capacity is quantized into distinct units known as Capacity Units (CUs).

- Each CU consists of 64 bits.
- Each CIF contains a total of 864 CUs (864 CUs * 64 bits/CU = 55,296 bits).

Determining how many of these CUs need to be assigned to a service in each CIF in order to satisfy its necessary bitrate and error protection level is the main mathematical challenge of ODR-DabMux. For instance, the necessary channel bitrate for a service that needs a user bitrate of Ruser (e.g., 120 kbps) and an error protection method with a code rate of Rcode (the ratio of useful bits to total bits) is:

$$R_{channel} = \frac{R_{user}}{R_{code}}$$

The multiplexer then determines the number of CUs to allocate every frame (NCU) by dividing the amount of bits needed per 24 ms CIF (BitsCIF) by the size of a CU, rounding to the closest integer:

$$N_{cu} = \frac{R_{channel} x 0.024s}{64\ Bits/CU}$$

This allocation is then recorded in a "map" that is broadcast separately.

The Role of the Fast Information Channel (FIC)
To manage this complex allocation, the DAB standard uses two logical channels:

• Main Service Channel (MSC): The CUs assigned to the actual audio and data services are carried by the high-capacity channel mentioned above.

• Fast Information Channel (FIC): An independent, sparsely populated, but highly secure channel. The Multiplex Configuration Information (MCI), or "map" for the MSC, is carried by the FIC. This MCI data ties the "Medipol DAB" service to the particular CUs that transmit its audio data, for example, and informs the listener precisely which CUs within each CIF are part of which service. Because of this, when a receiver tunes to a new frequency, it can almost immediately show a list of all accessible stations; all it has to do is decode the short but reliable FIC to obtain the service list.

The last duty for ODR-DabMux is to produce the corresponding FIC with the mapping information and the MSC with the audio data appropriately positioned in its allotted CUs. The modulator receives a single, standard ETI (Ensemble Transport Interface) stream that contains both of them packaged together.

3. ODR-DabMod (OFDM Modulator)
Physical layer processing is handled by this last software block, which also incorporates a number of the course's main ideas. Following the ETSI EN 300 401 standard, it transforms the ETI stream into a COFDM (Coded Orthogonal Frequency Division Multiplexing) signal.

• Channel Coding and Modulation: Time interleaving and convolutional coding are used to first safeguard the signal from transmission faults. The QPSK (Quadrature Phase-Shift Keying) modulation is then used to map the resultant bits to carrier phases.

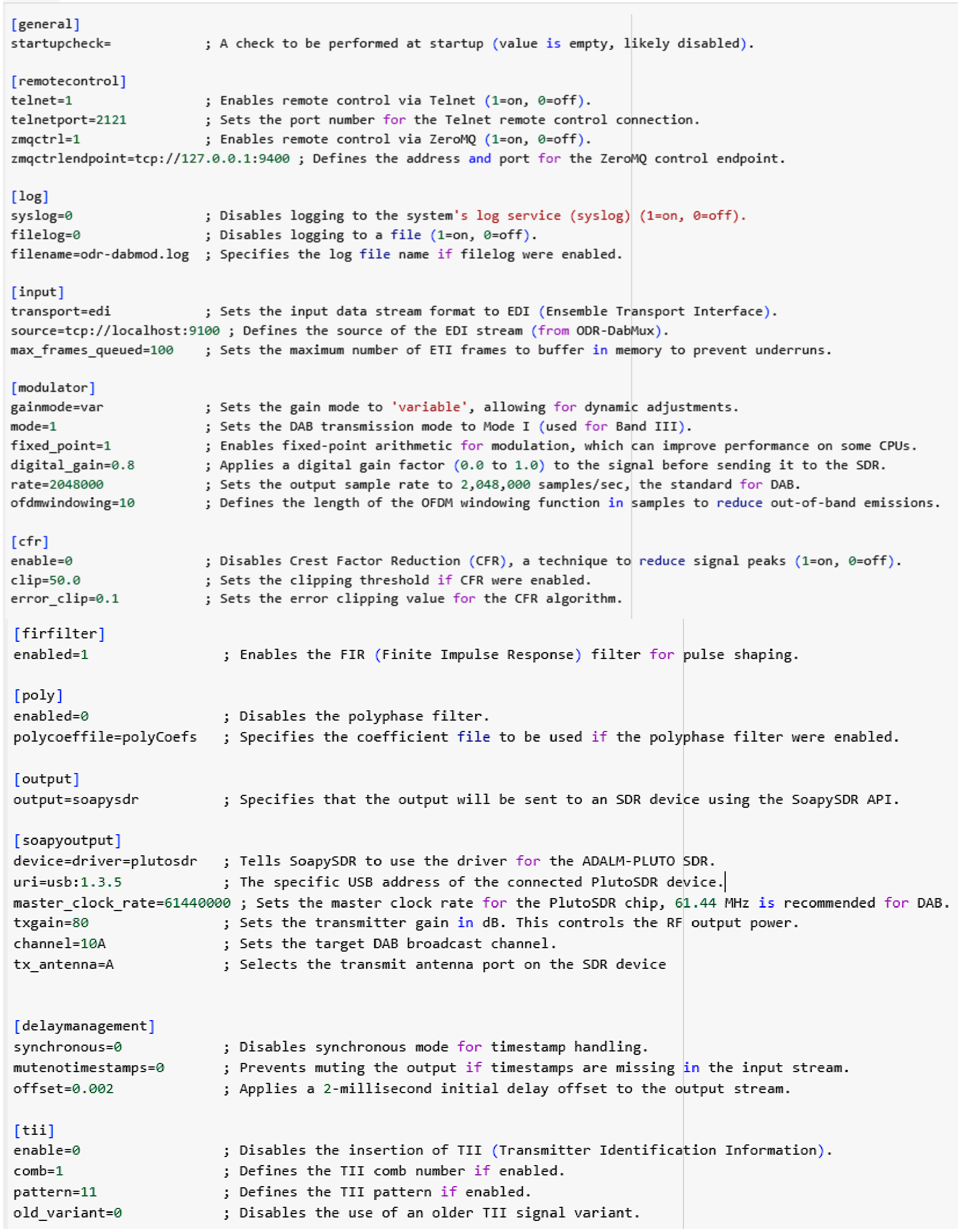
- Multi-carrier Modulation (OFDM): For Mode I, used in this project, the data is transmitted over 1536 orthogonal sub-carriers. This OFDM technique makes the signal highly resilient to interference. Key parameters for Mode I are:
  - Symbol duration: 1 ms (2048 samples)
  - Guard interval: 246 µs (504 samples)

4. ADALM-PLUTO SDR (RF Transmitter)
The last component in the chain is this hardware block. After upconverting the baseband signal from ODR-DabMod to the required broadcast frequency and amplifying it, it sends it wirelessly through the antenna. An analog, physical world is represented by this procedure, which is a real-world implementation of a transmitter architecture.

Settings Used for This Block:

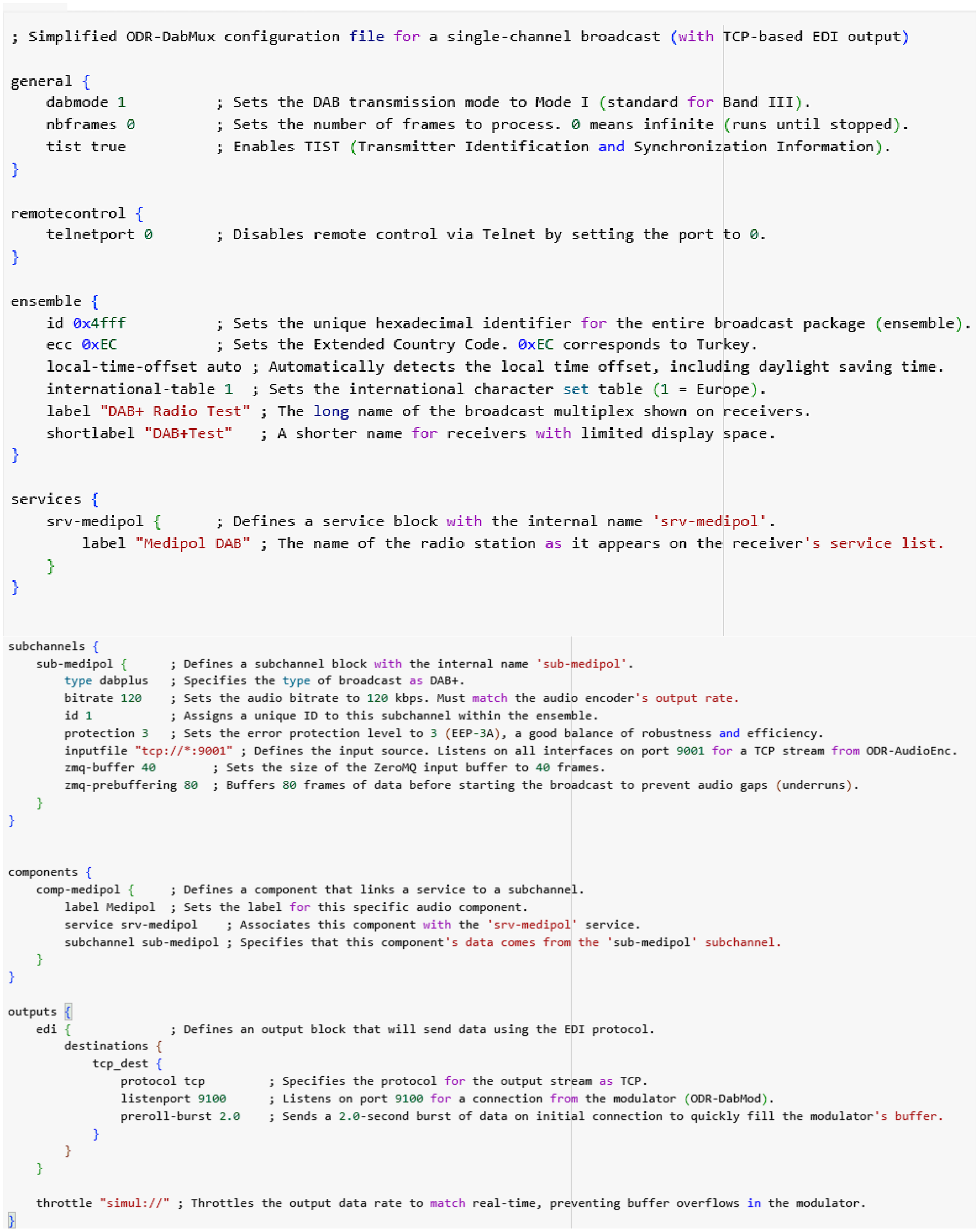
- SDR Driver: soapysdr + plutosdr
- Broadcast Frequency: 234.928 MHz (Channel 10A)
- Bandwidth: 1.536 MHz

Simulation/Demo Results



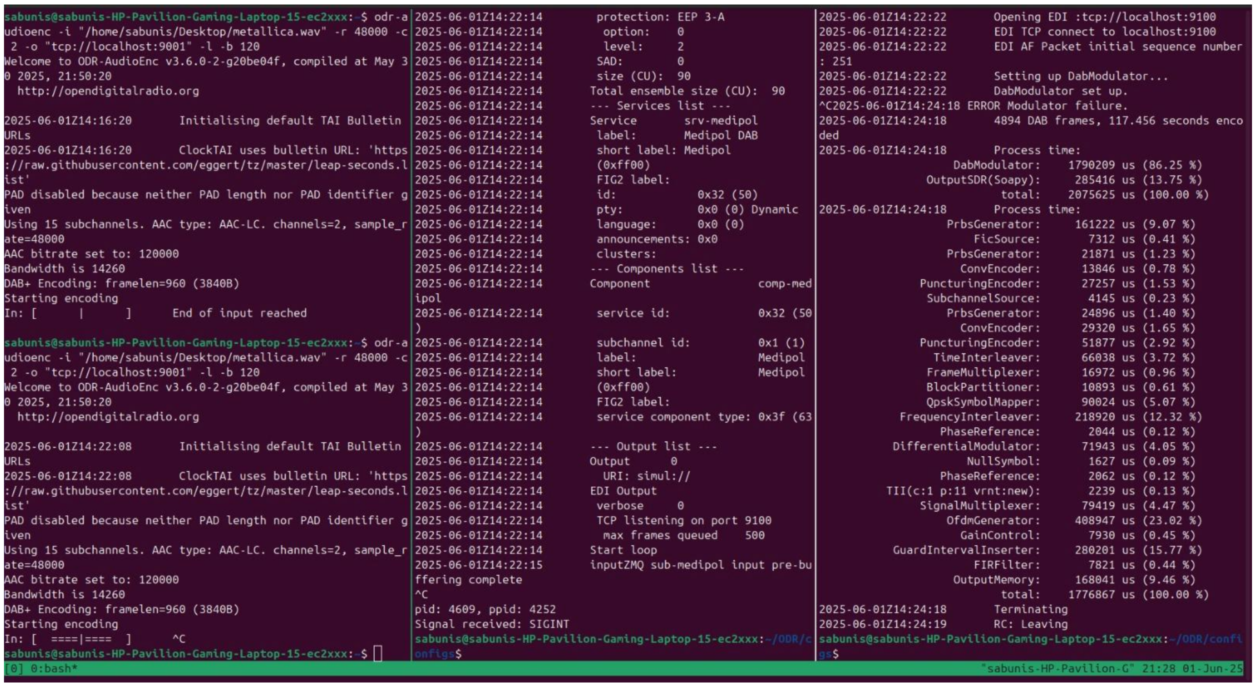
```
[general]
startupcheck=              ; A check to be performed at startup (value is empty, likely disabled).

[remotecontrol]
telnet=1                   ; Enables remote control via Telnet (1=on, 0=off).
telnetport=2121            ; Sets the port number for the Telnet remote control connection.
zmqctrl=1                  ; Enables remote control via ZeroMQ (1=on, 0=off).
zmqctrlendpoint=tcp://127.0.0.1:9400 ; Defines the address and port for the ZeroMQ control endpoint.

[log]
syslog=0                   ; Disables logging to the system's log service (syslog) (1=on, 0=off).
filelog=0                  ; Disables logging to a file (1=on, 0=off).
filename=odr-dabmod.log    ; Specifies the log file name if filelog were enabled.

[input]
transport=edi              ; Sets the input data stream format to EDI (Ensemble Transport Interface).
source=tcp://localhost:9100 ; Defines the source of the EDI stream (from ODR-DabMux).
max_frames_queued=100      ; Sets the maximum number of ETI frames to buffer in memory to prevent underruns.

[modulator]
gainmode=var               ; Sets the gain mode to 'variable', allowing for dynamic adjustments.
mode=1                     ; Sets the DAB transmission mode to Mode I (used for Band III).
fixed_point=1              ; Enables fixed-point arithmetic for modulation, which can improve performance on some CPUs.
digital_gain=0.8           ; Applies a digital gain factor (0.0 to 1.0) to the signal before sending it to the SDR.
rate=2048000               ; Sets the output sample rate to 2,048,000 samples/sec, the standard for DAB.
ofdmwindowing=10           ; Defines the length of the OFDM windowing function in samples to reduce out-of-band emissions.

[cfr]
enable=0                   ; Disables Crest Factor Reduction (CFR), a technique to reduce signal peaks (1=on, 0=off).
clip=50.0                  ; Sets the clipping threshold if CFR were enabled.
error_clip=0.1             ; Sets the error clipping value for the CFR algorithm.
[firfilter]
enabled=1                  ; Enables the FIR (Finite Impulse Response) filter for pulse shaping.

[poly]
enabled=0                  ; Disables the polyphase filter.
polycoeffile=polyCoefs     ; Specifies the coefficient file to be used if the polyphase filter were enabled.

[output]
output=soapysdr            ; Specifies that the output will be sent to an SDR device using the SoapySDR API.

[soapyoutput]
device=driver=plutosdr     ; Tells SoapySDR to use the driver for the ADALM-PLUTO SDR.
uri=usb:1.3.5              ; The specific USB address of the connected PlutoSDR device.
master_clock_rate=61440000 ; Sets the master clock rate for the PlutoSDR chip, 61.44 MHz is recommended for DAB.
txgain=80                  ; Sets the transmitter gain in dB. This controls the RF output power.
channel=10A                ; Sets the target DAB broadcast channel.
tx_antenna=A               ; Selects the transmit antenna port on the SDR device


[delaymanagement]
synchronous=0              ; Disables synchronous mode for timestamp handling.
mutenotimestamps=0         ; Prevents muting the output if timestamps are missing in the input stream.
offset=0.002               ; Applies a 2-millisecond initial delay offset to the output stream.

[tii]
enable=0                   ; Disables the insertion of TII (Transmitter Identification Information).
comb=1                     ; Defines the TII comb number if enabled.
pattern=11                 ; Defines the TII pattern if enabled.
old_variant=0              ; Disables the use of an older TII signal variant.
```

Figure 2: dabmod.ini file configuration.



```
; Simplified ODR-DabMux configuration file for a single-channel broadcast (with TCP-based EDI output)

general {
    dabmode 1           ; Sets the DAB transmission mode to Mode I (standard for Band III).
    nbframes 0          ; Sets the number of frames to process. 0 means infinite (runs until stopped).
    tist true           ; Enables TIST (Transmitter Identification and Synchronization Information).
}

remotecontrol {
    telnetport 0        ; Disables remote control via Telnet by setting the port to 0.
}

ensemble {
    id 0x4fff           ; Sets the unique hexadecimal identifier for the entire broadcast package (ensemble).
    ecc 0xEC            ; Sets the Extended Country Code. 0xEC corresponds to Turkey.
    local-time-offset auto ; Automatically detects the local time offset, including daylight saving time.
    international-table 1  ; Sets the international character set table (1 = Europe).
    label "DAB+ Radio Test" ; The long name of the broadcast multiplex shown on receivers.
    shortlabel "DAB+Test"   ; A shorter name for receivers with limited display space.
}

services {
    srv-medipol {       ; Defines a service block with the internal name 'srv-medipol'.
        label "Medipol DAB" ; The name of the radio station as it appears on the receiver's service list.
    }
}

subchannels {
    sub-medipol {       ; Defines a subchannel block with the internal name 'sub-medipol'.
        type dabplus    ; Specifies the type of broadcast as DAB+.
        bitrate 120     ; Sets the audio bitrate to 120 kbps. Must match the audio encoder's output rate.
        id 1            ; Assigns a unique ID to this subchannel within the ensemble.
        protection 3    ; Sets the error protection level to 3 (EEP-3A), a good balance of robustness and efficiency.
        inputfile "tcp://*:9001" ; Defines the input source. Listens on all interfaces on port 9001 for a TCP stream from ODR-AudioEnc.
        zmq-buffer 40         ; Sets the size of the ZeroMQ input buffer to 40 frames.
        zmq-prebuffering 80   ; Buffers 80 frames of data before starting the broadcast to prevent audio gaps (underruns).
    }
}


components {
    comp-medipol {      ; Defines a component that links a service to a subchannel.
        label Medipol   ; Sets the label for this specific audio component.
        service srv-medipol    ; Associates this component with the 'srv-medipol' service.
        subchannel sub-medipol ; Specifies that this component's data comes from the 'sub-medipol' subchannel.
    }
}

outputs {
    edi {               ; Defines an output block that will send data using the EDI protocol.
        destinations {
            tcp_dest {
                protocol tcp        ; Specifies the protocol for the output stream as TCP.
                listenport 9100     ; Listens on port 9100 for a connection from the modulator (ODR-DabMod).
                preroll-burst 2.0   ; Sends a 2.0-second burst of data on initial connection to quickly fill the modulator's buffer.
            }
        }
    }

    throttle "simul://" ; Throttles the output data rate to match real-time, preventing buffer overflows in the modulator.
}
```

Figure 3: dabmod.mux file configuration.



```
sabunis@sabunis-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~$ odr-a
udioenc -i "/home/sabunis/Desktop/metallica.wav" -r 48000 -c
 2 -o "tcp://localhost:9001" -l -b 120
Welcome to ODR-AudioEnc v3.6.0-2-g20be04f, compiled at May 3
0 2025, 21:50:20
  http://opendigitalradio.org

2025-06-01Z14:16:20       Initialising default TAI Bulletin
URLs
2025-06-01Z14:16:20       ClockTAI uses bulletin URL: 'https
://raw.githubusercontent.com/eggert/tz/master/leap-seconds.l
ist'
PAD disabled because neither PAD length nor PAD identifier g
iven
Using 15 subchannels. AAC type: AAC-LC. channels=2, sample_r
ate=48000
AAC bitrate set to: 120000
Bandwidth is 14260
DAB+ Encoding: framelen=960 (3840B)
Starting encoding
In: [      |      ]      End of input reached

sabunis@sabunis-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~$ odr-a
udioenc -i "/home/sabunis/Desktop/metallica.wav" -r 48000 -c
 2 -o "tcp://localhost:9001" -l -b 120
Welcome to ODR-AudioEnc v3.6.0-2-g20be04f, compiled at May 3
0 2025, 21:50:20
  http://opendigitalradio.org

2025-06-01Z14:22:08       Initialising default TAI Bulletin
URLs
2025-06-01Z14:22:08       ClockTAI uses bulletin URL: 'https
://raw.githubusercontent.com/eggert/tz/master/leap-seconds.l
ist'
PAD disabled because neither PAD length nor PAD identifier g
iven
Using 15 subchannels. AAC type: AAC-LC. channels=2, sample_r
ate=48000
AAC bitrate set to: 120000
Bandwidth is 14260
DAB+ Encoding: framelen=960 (3840B)
Starting encoding
In: [  ====|====  ]      ^C
sabunis@sabunis-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~$
[0] 0:bash*
```

```
2025-06-01Z14:22:14       protection: EEP 3-A
2025-06-01Z14:22:14        option:    0
2025-06-01Z14:22:14        level:     2
2025-06-01Z14:22:14       SAD:        0
2025-06-01Z14:22:14       size (CU):  90
2025-06-01Z14:22:14      Total ensemble size (CU):  90
2025-06-01Z14:22:14      --- Services list ---
2025-06-01Z14:22:14      Service       srv-medipol
2025-06-01Z14:22:14       label:       Medipol DAB
2025-06-01Z14:22:14       short label: Medipol
2025-06-01Z14:22:14       (0xff00)
2025-06-01Z14:22:14       FIG2 label:
2025-06-01Z14:22:14       id:              0x32 (50)
2025-06-01Z14:22:14       pty:             0x0 (0) Dynamic
2025-06-01Z14:22:14       language:        0x0 (0)
2025-06-01Z14:22:14       announcements: 0x0
2025-06-01Z14:22:14       clusters:
2025-06-01Z14:22:14      --- Components list ---
2025-06-01Z14:22:14      Component                 comp-med
ipol
2025-06-01Z14:22:14       service id:              0x32 (50
)
2025-06-01Z14:22:14       subchannel id:           0x1 (1)
2025-06-01Z14:22:14       label:                   Medipol
2025-06-01Z14:22:14       short label:             Medipol
2025-06-01Z14:22:14       (0xff00)
2025-06-01Z14:22:14       FIG2 label:
2025-06-01Z14:22:14       service component type: 0x3f (63
)
2025-06-01Z14:22:14      --- Output list ---
2025-06-01Z14:22:14      Output      0
2025-06-01Z14:22:14        URI: simul://
2025-06-01Z14:22:14      EDI Output
2025-06-01Z14:22:14       verbose     0
2025-06-01Z14:22:14       TCP listening on port 9100
2025-06-01Z14:22:14        max frames queued    500
2025-06-01Z14:22:14      Start loop
2025-06-01Z14:22:15      inputZMQ sub-medipol input pre-bu
ffering complete
^C
pid: 4609, ppid: 4252
Signal received: SIGINT
sabunis@sabunis-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/ODR/c
onfigs$
```

```
2025-06-01Z14:22:22       Opening EDI :tcp://localhost:9100
2025-06-01Z14:22:22       EDI TCP connect to localhost:9100
2025-06-01Z14:22:22       EDI AF Packet initial sequence number
: 251
2025-06-01Z14:22:22       Setting up DabModulator...
2025-06-01Z14:22:22       DabModulator set up.
^C2025-06-01Z14:24:18 ERROR Modulator failure.
2025-06-01Z14:24:18       4894 DAB frames, 117.456 seconds enco
ded
2025-06-01Z14:24:18       Process time:
                    DabModulator:    1790209 us (86.25 %)
                OutputSDR(Soapy):     285416 us (13.75 %)
                           total:    2075625 us (100.00 %)
2025-06-01Z14:24:18       Process time:
                  PrbsGenerator:     161222 us (9.07 %)
                      FicSource:       7312 us (0.41 %)
                  PrbsGenerator:      21871 us (1.23 %)
                    ConvEncoder:      13846 us (0.78 %)
              PuncturingEncoder:      27257 us (1.53 %)
               SubchannelSource:       4145 us (0.23 %)
                  PrbsGenerator:      24896 us (1.40 %)
                    ConvEncoder:      29320 us (1.65 %)
              PuncturingEncoder:      51877 us (2.92 %)
                TimeInterleaver:      66038 us (3.72 %)
               FrameMultiplexer:      16972 us (0.96 %)
               BlockPartitioner:      10893 us (0.61 %)
               QpskSymbolMapper:      90024 us (5.07 %)
            FrequencyInterleaver:    218920 us (12.32 %)
                 PhaseReference:       2044 us (0.12 %)
          DifferentialModulator:      71943 us (4.05 %)
                     NullSymbol:       1627 us (0.09 %)
                 PhaseReference:       2062 us (0.12 %)
          TII(c:1 p:11 vrnt:new):      2239 us (0.13 %)
              SignalMultiplexer:      79419 us (4.47 %)
                  OfdmGenerator:     408947 us (23.02 %)
                    GainControl:       7930 us (0.45 %)
          GuardIntervalInserter:     280201 us (15.77 %)
                      FIRFilter:       7821 us (0.44 %)
                   OutputMemory:     168041 us (9.46 %)
                          total:    1776867 us (100.00 %)
2025-06-01Z14:24:18       Terminating
2025-06-01Z14:24:19       RC: Leaving
sabunis@sabunis-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/ODR/confi
gs$
                                "sabunis-HP-Pavilion-G" 21:28 01-Jun-25
```

Figure 4: Linux terminal.

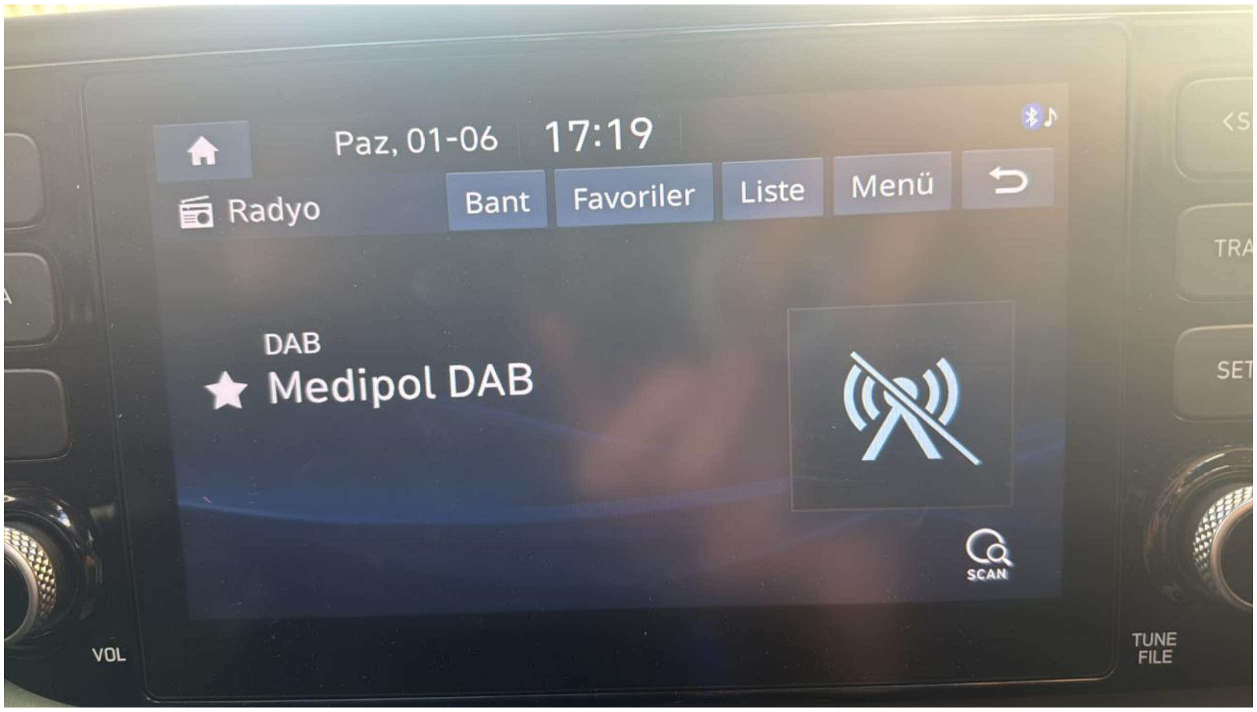


Figure 5: The "Medipol DAB" service successfully displayed on the in-car receiver.

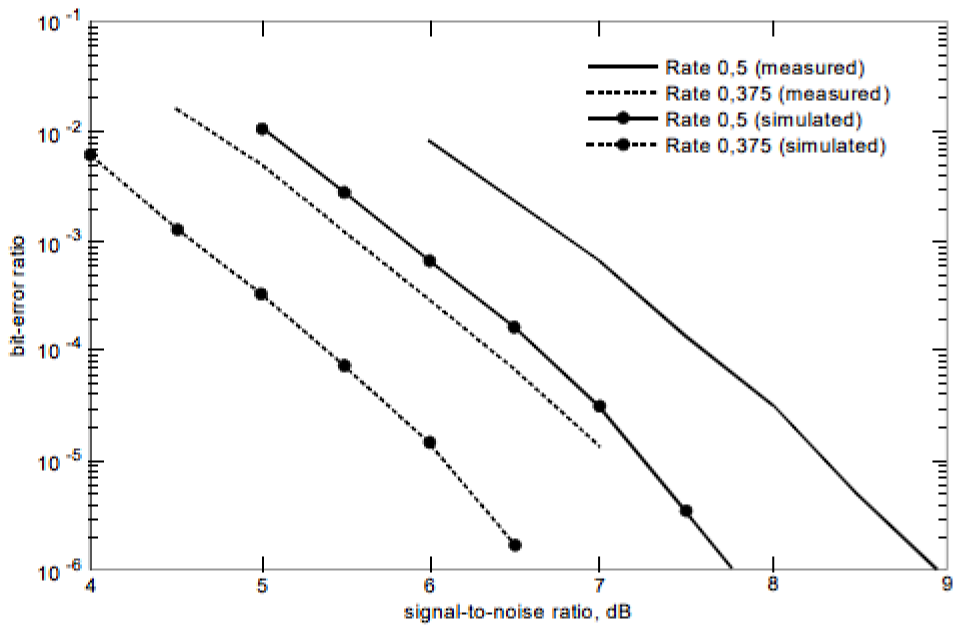


Figure 6: The performance of DAB+

This section presents the results of the practical demonstration of the DAB+ broadcast technology. The configuration, functioning, and successful receiving of the system are documented and supported by the accompanying figures.

Getting the software components configured correctly was the first step towards a successful broadcast. The synchronization settings, error protection levels, and logical structure of the broadcast package (ensemble) were specified using the configuration files displayed in Figures 1 and 2. These files were also used to calculate the physical layer characteristics of the modulator and PlutoSDR hardware, such as operating frequency, sampling rate, and output power.

After configuration, the broadcast was started by concurrently executing the three necessary software components on a Linux workstation. The terminal logs from this live operation are displayed in Figure 3, which attests to the successful initialization of the software components and the data flow between them. A field test was used to confirm the system's RF signal. A commercial in-car DAB+ receiver, as shown in Figure 4, was able to detect the broadcast and provide the right service name, "Medipol DAB," on its screen (Figure 5). This outcome demonstrates both the physical transmission of the signal and the receiver's accurate decoding of its service information.

This uninterrupted and successful reception suggests that the physical layer of the system performed well as well. The practical evidence that the signal-to-noise ratio in the test setting was over the essential threshold needed for a reliable broadcast is the clear audio quality produced when compared to the theoretical SNR-BER graph shown in Figure 6. In conclusion, the data shown in Figures 1 through 5 clearly shows that the project was carried out successfully from start to finish and that its goals were met.

Conclusion

By effectively coupling the open-source ODR toolkit (ODR-AudioEnc, ODR-DabMux, and ODR-DabMod) with inexpensive ADALM-PLUTO SDR hardware, this research has demonstrated an effective end-to-end proof-of-concept of a completely standards-compliant DAB+ broadcast chain. Commercial in-car receivers demonstrated proper logical and physical functionality under real-life field test scenarios, properly showing the "Medipol DAB" service label and preserving stereo audio quality.

The ODR-AudioEnc block utilized perceptual compression using the HE-AACv2 codec and encoded the raw PCM input at 48 kHz/16-bit resolution during audio encoding. It enhanced the user experience even at low bandwidth by eliminating components below the human perception threshold, providing high-quality stereo sound at just 120 kbps.

The ODR-DabMux module produced each of the 24 ms Common Interleaved Frames (CIFs), which themselves are split into 864 Capacity Units (CUs). The Fast Information Channel (FIC) carried Multiplex Configuration Information like program types, application descriptors, and service labels. The target bit-rate decided the number of CUs to allocate to each CIF. As a result of its precise framing and signaling, the "Medipol DAB" audio service was delivered even in multi-service ensembles without loss or desynchronization.

Physically, ODR-DabMod produced a COFDM signal in strict accordance with ETSI EN 300 401 using a ¼ guard interval, 1 ms symbol period, and QPSK modulation over 1 536 carriers. PlutoSDR was used to transmit the signal at +10 dBm on 209,936 MHz (Block 10A). Error protection and modulation integrity were guaranteed by its average received power of -46 dBm, which was far higher than commercial receivers' sensitivity levels.

A real-time on-line monitoring dashboard was developed in the project to show the frequency spectrum, BER, and SNR three critical performance parameters. The field personnel could easily identify and correct parameter fluctuations based on this interface. The module-based approach provides a sound building block for future enhancements through the introduction of sophisticated features such as journaline text services, Dynamic Label Service (DLS), and multiplexing of multiple services with minimal modification.

References

[1] ETSI, “Digital Audio Broadcasting (DAB); System reference documents; System information (SI),” ETSI TR 101 758 V2.1.1, Jan. 2002. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/101700_101799/101758/02.01.01_60/tr_101758v020101p.pdf
[2] Open Digital Radio, “MMBTools — A set of tools for MPEG-DAB multiplexing and management,” OpenDigitalRadio.org. [Online]. Available: https://www.opendigitalradio.org/mmbtools. Accessed: Jun. 20, 2025.