

# 「Web ベース放送受信システム」における 動画再生機能の試作と検証

佐藤辰哉<sup>1</sup> 遠藤大礎<sup>1</sup> 大亦寿之<sup>1</sup> 松村欣司<sup>1</sup> 藤沢寛<sup>1</sup>

**概要:** 近年, インターネット接続機能を備えたスマートテレビの普及が進んでいる. 多くのスマートテレビには, 放送の受信機能に加え, 動画配信など多様なインターネット (ネット) サービスを提供するための Web ブラウザが搭載されている. Web は順次更新される標準技術であり, 汎用的な Web ブラウザを採用すると, 標準技術の進化に合わせたサービスの発展が容易になる. 一方, ハイブリッドキャストなどの放送通信連携システムに対応した受信機も, アプリケーション実行環境として Web ブラウザを用いている. 放送信号に含まれる映像や音声, 各種データなどの放送リソースを参照・提示するために, 独自の拡張を行っている. そのため, 放送とネット動画で異なる HTML 要素を指定する必要があり, 単一の動画再生機能で双方のコンテンツを再生できないという課題がある. そこで筆者らは, 放送とネットによるコンテンツの提供・管理を容易にすることを目的に, 伝送路ごとに個別の機能を必要としない単一の Web アプリケーションでコンテンツを再生可能な「Web ベース放送受信システム」を設計した. また, 本システムにおいて, 地上デジタル放送の信号の参照・提示が可能な動画再生機能を試作し, 動作検証によりその妥当性評価を行った.

**キーワード:** 放送通信連携, 受信機, Web, Media Source Extensions

## Prototype of Video Playback Function for Web-based Broadcasting Receiving System

TATSUYA SATO<sup>†1</sup> HIROKI ENDO<sup>†1</sup>  
HISAYUKI OHMATA<sup>†1</sup> KINJI MATSUMURA<sup>†1</sup> HIROSHI FUJISAWA<sup>†1</sup>

**Abstract:** In recent years, smart TVs equipped with Internet connectivity have become increasingly popular. Many smart TVs are equipped with a Web browser to provide various Internet services, such as video distribution, in addition to their broadcasting reception functions. On the other hand, broadcasting and communication cooperative systems such as Hybridcast also use Web browsers. However, in order to reference and present broadcasting resources such as video, audio, and various data included in the broadcasting signal, broadcasting has its own extensions that are not part of the Web standard technology, and because different elements need to be specified for broadcasting and Internet video. Because it is necessary to specify different elements for broadcast and online video, a single video playback function cannot playback the content. Therefore, the authors designed a "Web-based Broadcasting Receiving System" that enables content playback with a single Web application (common Web application for broadcasting and communications) that does not require separate functions for each transmission channel, in order to facilitate the provision and management of content for broadcasting and the Internet. The authors also built a prototype of a Web-based receiving system compatible with current broadcast signals and verified the operation of the video playback function.

**Keywords:** Integrated broadcast-broadband system, Broadcasting Receiver, Web, Media Source Extensions

### 1. はじめに

日本におけるデジタル放送は, 2000 年の BS デジタル放送の開始から約 20 年が経過した. 対応受信機の普及も約 95% となっており, 受信機はどの家庭にもある一般的なデバイスとなっている. とくにこの数年でインターネット接続機能を備えたスマートテレビの普及が進み, これまでのデジタル放送の受信機能に加え, 動画配信など多様なインターネット (ネット) サービスを一台の受信機で利用することができるようになってきた. また, 放送コンテンツとネットコンテンツの連携が可能な放送通信連携サービスの標準規格として, 日本のハイブリッドキャスト[1]や欧州の

HbbTV[2]が挙げられる. 国内のハイブリッドキャスト対応受信機の累計出荷台数も 2,000 万台を超えるなど, サービスの普及が進んでいる[3].

このように, 受信機において, 放送だけでなくネットサービスやコンテンツを誰もが利用できる環境が当たり前になる中で, 筆者らは Web 技術をベースとした放送サービスの基盤技術である「Web ベース放送プラットフォーム」[4]の検討を進めている. このプラットフォームは, Web 技術を活用し, コンテンツや視聴環境などに関する様々なデータが連携することで放送サービスが生活のすみずみまで届くことを目指したものである. 本プラットフォームにお

<sup>1</sup> 日本放送協会  
Japan Broadcasting Corporation

る受信機が「Web ベース放送受信システム」である。Web の標準技術の進化に追従した機能のアップデートや様々な機能の連携動作を容易とすることで、より豊かな放送サービスの提供を目指して開発を進めている。

本稿では、この Web ベース放送受信システムのアーキテクチャの設計と、最も重要な要素の一つである動画再生機能の試作と検証を行ったので報告する。本アーキテクチャは、Web アプリケーション（アプリ）とその実行環境である汎用ブラウザとによるプレイヤー部と、放送と通信の伝送路を意識せずブラウザから参照可能とする伝送路抽象化機能部の2つの機能部により構成する。この構成により、放送とネット動画を同じ仕組みで提示可能できる設計とした。また、動画再生機能については、日本のデジタル放送方式である ISDB-T (Integrated Services Digital Broadcasting - Terrestrial) 方式の信号に対応するための設計と試作を行った。実装とその動作検証により、汎用ブラウザの軽微な改修だけで、ISDB-T の放送信号を再生できることを確認した。

## 2. 関連技術

一台の受信機で放送とネットの両方のサービスを利用できるようにするためには、受信機アーキテクチャとサービスの切り替え方式の2つの要素技術を考慮する必要がある。

図1に日本の標準的なデジタル放送対応受信機のアーキテクチャと放送とネット動画の切り替え方式を示す。放送とネット動画の連携は、これまでデータ放送やハイブリッドキャストのアプリを使って用いて実現され、放送事業者による放送通信連携サービスの一部として提供されてきた。データ放送やサービス開始当初のハイブリッドキャストでは、各アプリと専用の動画プレイヤーを遷移させることで、放送と動画の切り替えを可能とした（方式1）。2016年には、ハイブリッドキャストのブラウザ仕様を規定する

IPTV フォーラムの標準規格[5]において W3C 標準の Media Source Extensions (MSE) [6]を採用し、ハイブリッドキャストのアプリで動画再生が可能となった。この MSE は、Web アプリにおいてプラグインを利用することなく標準の HTML 要素である video 要素を用いて動画のファイル再生およびストリーム再生を可能とする API であり、現在普及しているほぼ全ての汎用ブラウザにおいて対応している。ハイブリッドキャストでは object 要素を用いて放送信号を指定し、汎用的なブラウザを放送用に拡張することで放送の映像・音声の提示や各種データの参照を可能としている。放送と動画の遷移は、この object 要素と video 要素を切り替えることにより実現している（方式2）。

しかし、これらの方式に対応する受信機のアーキテクチャでは、アプリの実行環境である Web ブラウザに対し放送やネット動画と連携するための拡張が必要となる。また、放送とネットを切り替えるアプリはコンテンツの伝送路を意識して制作する必要がある。方式1では、データ放送やハイブリッドキャストと専用の動画プレイヤーのそれぞれにおいて相互に遷移するためのアプリの実装が必要である。方式2は、単一のアプリで切り替えが可能であるが、放送と動画で異なる要素を指定する必要がある。

一方、単一のユーザーインターフェース上で放送とネットのコンテンツを提供可能とする方式として、次世代放送方式である DVB-I[7]が挙げられる。また日本においても同様の技術研究[8]がある。これらの方式では現行の放送方式を維持しつつ、アプリケーションレイヤーで放送とネットを判別することで、伝送路に依存しないサービス提供が可能となっている。しかし、放送と通信のコンテンツはアプリケーションの下位レイヤーでは全く異なる仕組みが必要であり、受信機実装においても両者を意識する必要がある。また、伝送レイヤーで放送とネットを共通化する方式として ATSC3.0[9]がある。アプリケーションより下位レイヤーで放送とネットの区別をなくすため、アプリケーションからは伝送路を意識せずに両者を再生することが可能である。

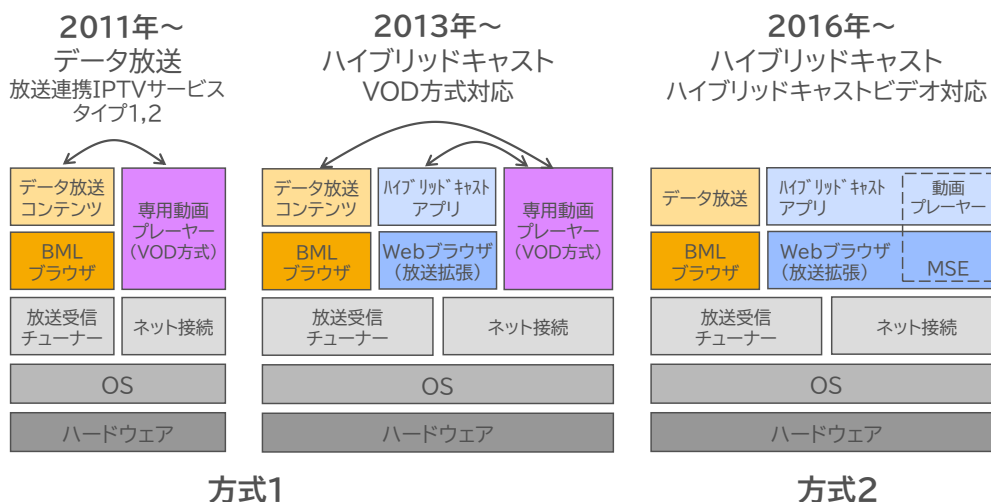


図1 放送とネット動画の切り替え方式

しかし、ATSC3.0は米国を中心に検討されている放送方式であり、この方式を日本で適用するための検討はなされていない。

筆者らは、これまで放送とネットのコンテンツが共通の仕組みで再生される受信機アーキテクチャを提案してきた[10]。しかし、文献[10]においては、プレイヤー部と伝送路抽象化機能部を物理的に分離するモデルを中心に検討を行い、同一筐体としての試作や検証は行っていなかった。また、放送信号をMPEG-DASH方式に変換してWebブラウザで再生する方式としたため、受信した放送信号をトランスコードする機能が受信機に必要であった。

### 3. Web ベース受信システムの提案

2章で述べた課題を解決する受信機として、単一のWebアプリである放送通信共通Webアプリを用いて放送とネット動画を再生可能とし、一つの筐体で構成する「Webベース放送受信システム」を提案する。

#### 3.1 設計指針

今回の設計にあたってはこれまでの基本設計における検証に基づき以下の通り3つの指針を定めた。

##### 指針1: プレイヤー部の汎用ブラウザを前提としたWebアプリとしての実装

プレイヤー部は放送連携機能を拡張したブラウザではなく、一般的に使用される汎用ブラウザにより構成する。また動画再生はこのブラウザ上で動作するWebアプリとして実装する。これによりWeb標準技術に追従したアップデートが容易になり、プレイヤー部のメンテナンスや拡張もしやすくなる。

##### 指針2: 受信機負荷と動画再生遅延の軽減

これまでの検証から、受信機の実現性を高めるために伝送路抽象化部における受信機負荷を軽減する必要があることが分かった。また放送の速報性・同報性を損なわないよう動画再生遅延を軽減することも必要である。

##### 指針3: 論理的に分離するプレイヤー部、伝送路抽象化機能部の一体型モデル

Webブラウザやミドルウェア、伝送路の方式が、それぞれ互いに依存することなく機能の更新を容易に可能とするため、プレイヤー部と伝送路抽象化機能部は論理的に分離する構成として設計する。今回は現行の受信機に沿う形態として同一筐体にプレイヤー部と伝送路抽象化機能部を実装することとした。

#### 3.2 提案システムの構成

図4にWebベース受信システムの構成図を示す。Webベース放送受信システムの基本設計では、Webブラウザ上のWebアプリケーションでコンテンツを参照・提示するプレイヤー部と受信するコンテンツの伝送路が放送かネットかを問わずにブラウザから扱えるようにするためのミドルウェアである伝送路抽象化機能部から構成される。

#### 3.3 プレイヤー部

指針1に基づき、プレイヤー部は汎用ブラウザと放送通信共通Webアプリにより構成する。またWebブラウザと伝送路抽象化機能部との間で放送リソースの取得・制御を行うためのWebAPIを設けることとした。WebAPIを提供する機能として放送リソース取得APIと制御APIを備える。放送通信共通Webアプリは、このWebAPIを実行することで動画の再生や選局動作などのチューナー相当の機能を実現する。また動画の再生にはMSEを用いることとし

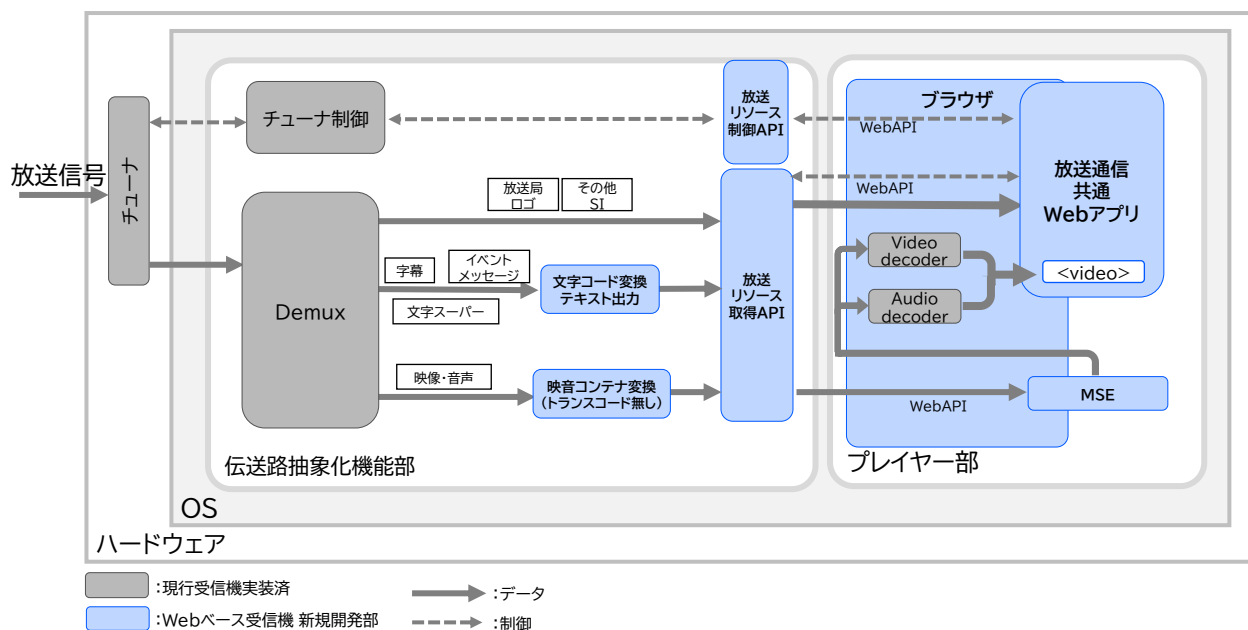


図2 Web ベース受信システムの構成

た。

### 3.4 伝送路抽象化機能部

これまでの検証[10]においては ISDB-T 方式の放送信号を、汎用ブラウザで再生可能な MP4 などの動画フォーマットにトランスコードし、動画再生を行っていた。しかし動画のトランスコードは受信機への負荷が大きく、遅延の要因となることが考えられる。したがって今回は指針 2 の対応として、放送信号をトランスコードせずに MSE において再生可能な形に変換することとした。また放送信号の再生に対応するため、汎用ブラウザの改修を行った。改修の詳細は 4 章で述べる。

### 3.5 放送再生フロー

この構成における放送の再生フローについて説明する。チューナーで受信した放送信号はデマルチプレクサによって、映像・音声・字幕・文字スーパーなどの要素ごとに分離される。その後それぞれ Web ブラウザで再生するために、動画のコンテナ形式や文字コードの変換など必要な処理を行う。そして Web アプリからの要求に応じて WebAPI を通して必要な放送信号が取得されプレイヤーに提示される。

## 4. 試作

ISDB-T 方式の放送に対応する図 5 に示す構成の Web ベース放送受信システムの試作を行った。指針 3 に基づき、同一筐体内にプレイヤー部と伝送路抽象化機能部を実装した。ハードウェアには一般的な受信機としての組み込みを想定し、シングルボードコンピュータの Raspberry Pi 4 ModelB (8GB) を使用した。また OS は Linux ベースの

Raspberry Pi OS (64bit バージョン 11 bullseye) とした。プレイヤー部と伝送路抽象化機能部の実装の詳細を以下に示す。

### 4.1 プレイヤー部の試作

放送やネット動画の再生を担うプレイヤー部は Web ブラウザと放送通信共通 Web アプリから構成される。今回現行放送再生可能なブラウザと Web ベース放送受信システム検証用 Web アプリ (検証用 Web アプリ) を試作した。

#### 4.1.1 Web ブラウザの試作

プレイヤー部の Web ブラウザは Chromium[11]に後述する軽微な改修を加えたものを用いた。Chromium はオープンソースとして提供されるブラウザであり、Google Chrome[12]や Microsoft Edge[13]などのブラウザのコアにも採用されていることから汎用ブラウザとして用いることとした。また放送信号をトランスコードせずにブラウザの MSE を用いて再生できるようにした。

Chromium の動画再生機能では、ISDB-T 方式の映像フォーマットである MPEG2-TS やその符号化方式である MPEG2-Video に対応した映像を再生できない。そこで放送信号の再生に対応するため、以下の 3 項目について Chromium の改修を行った。

#### ① MPEG2-TS および MPEG2-Video の MIME type/codec 有効化

Chromium のデフォルト状態では、MPEG2-TS および MPEG2-Video の MIME Type に非対応である。MIME type はサーバ・クライアント間の通信においてデータの形式を指定するものであり、正しく設定されないと送受信するデータに対し、適切な処理を行うことができない。本件においては、ブラウザがファイルの認識ができないため、MSE の

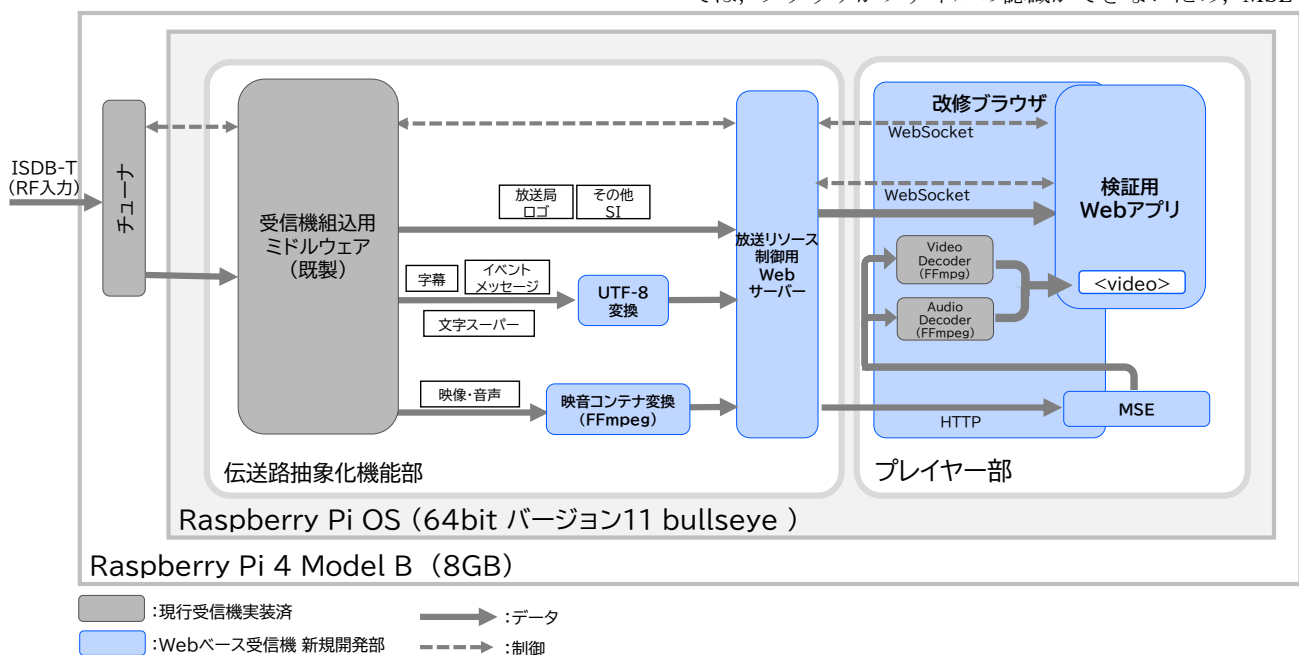


図 3 試作システムの構成

ソース指定に失敗する。このため今回、Chromium の MIME Type サポート判定処理に対し MPEG2-TS および MPEG2-Video の MIME type / codec 対応を追加し、サポートを有効化した。

### ② MPEG2-TS パーサ機能有効化

MPEG2-TS のパーサ機能が無効化されているため、MPEG2-TS コンテナがサポートされていない。MPEG2-TS から映像・音声データを取り出すためのデマルチプレクサ機能に対応するため“MPEG2TS\_STREAM\_PARSER”を有効化した。また Chromium の動画再生用 API に MPEG2-Video のデータを渡すため、MPEG2-TS から映像のフレームごとに MPEG2-Video を切り出す処理を追加実装した。

### ③ MPEG2-Video デコード機能有効化

Chromium では動画ファイルのデコードに FFmpeg[14]を利用している。Chromium のデフォルト状態においてはこの FFmpeg の MPEG2-TS デコード機能が無効化されているため、これを有効化した。

#### 4.1.2 検証用 Web アプリの試作

改修したブラウザ上で動作する検証用 Web アプリを試作した。このアプリは放送リソース取得・制御 API を実行することができ、Web ブラウザから伝送路抽象化機能部を制御して提案システムの挙動を検証することができる。また放送信号を入力し、MSE を用いた動画再生や字幕、イベントメッセージの受信内容を確認することが可能である。

#### 4.2 伝送路抽象化機能部の試作

伝送路抽象化機能部として市販テレビに組み込まれている SDK (Software Development Kit) を組み込むことで放送受信機としての機能を動作させた。この SDK は放送信号のデマルチプレクサ機能や文字コード変換等も担う。また放送信号を MSE における動作に適した形式に変換するた

め FFmpeg(バージョン 5.0)を使用した。そしてプレイヤー部とのインターフェースとなる WebAPI を実装し、伝送路抽象化機能部を構成した。

## 5. 検証・考察

### 5.1 動画再生機能の検証

4章で述べた試作システムにおいて検証用 Web アプリを用いて動画再生機能の検証を行った。なお動画形式については今回の検証対象外としており、今回の検証においては簡単のため放送信号を MSE に利用できる HLS (HTTP Live Streaming) に近い形式で送出することとした。検証用 Web アプリにおける検証の様子を図 4 に示す。画面右上が MSE を用いた動画再生部となっており、検証により改修ブラウザにおいて放送信号が再生可能であることが確認できた。

### 5.2 考察

今回の試作により、放送信号が汎用ブラウザで MSE を用いて再生可能となることが確認できた。このアーキテクチャを用いると、MSE における再生リソースの指定を変えるだけで、再生コンテンツを変更することができる。つまり放送とネット動画の切り替えを MSE のパラメータを変更するだけで可能となり、単一プレイヤーでの切り替えの実現が可能であることを示した。

また今回の改修で必要となった作業は、4.1.1 に示した通り既に備わっているブラウザ機能の有効化により対応が可能であった。つまりブラウザの大規模なソースコードの書き換えや追加改修を必要とせず放送信号の再生が可能になることがわかった。これにより受信機のプレイヤー部として汎用ブラウザを用いる構成の実現可能性を確認することができた。

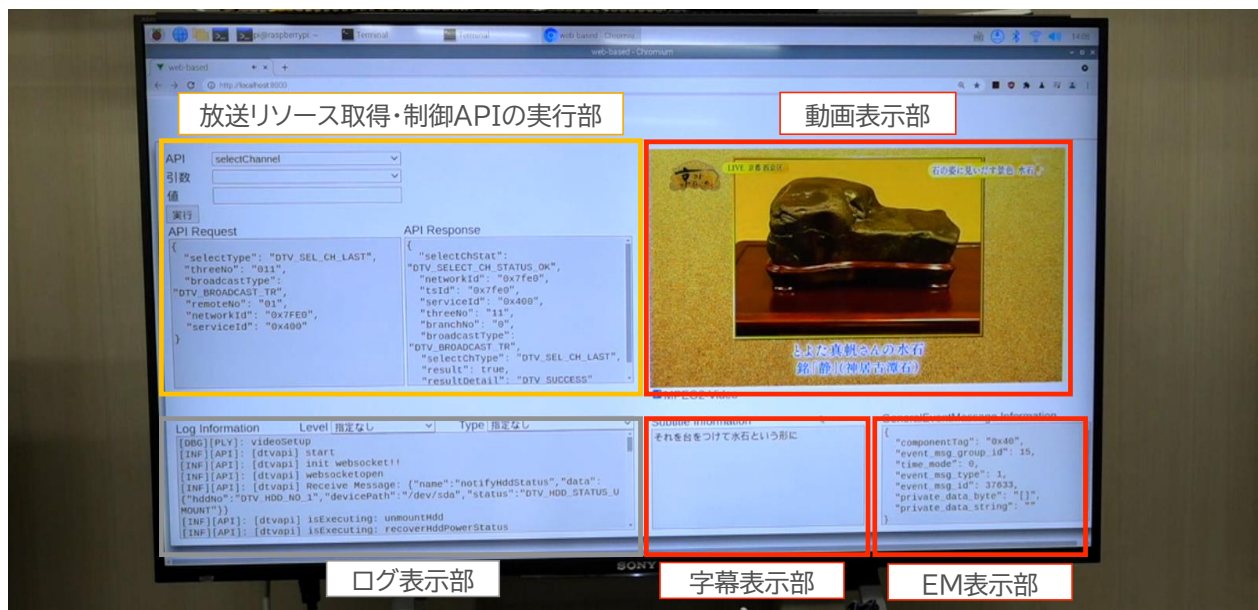


図 4 試作システムの動作例

一方、本試作における放送信号の受信から提示までの時間を市販のデジタル放送対応受信機と比較すると、約6秒の遅延があることを確認した。動画のHLS形式へのコンテンツ変換や再生時のバッファリングが原因と考えられる。放送において同報性や速報性は重要な要素であることから、遅延の抑制については今後の課題として検討を進めたい。

## 6. まとめ

本稿では放送通信共通 Web アプリによる放送とネット動画の再生を可能とする受信システムの実現を目的に Web ベース放送受信システムの設計と動画再生機能の検証を行った。受信システムの設計においては汎用ブラウザにおける Web アプリでの再生や放送信号をトランスコードしない方式とするなどの指針を基にした構成を提案した。また MSE を用いて放送信号を再生するために汎用ブラウザの改修を行った。結果として軽微な改修により放送再生が可能となった。これにより提案する Web ベース放送受信システムの実現可能性を示した。

今後は本構成におけるプレイヤー部と伝送路抽象化機能部のインターフェースとなる WebAPI の詳細設計や放送リソースへのアクセス制御手法などについて検討を進める。

## 参考文献

- [1] “Hybridcast”, <https://www.iptvforum.jp/hybridcast/>
- [2] “HbbTV”, <https://www.hbbtv.org/>
- [3] “JEITA 2022 年民生用電子機器国内出荷統計”, <https://www.jeita.or.jp/japanese/stat/shipment/2022/index2.htm>
- [4] “Web ベース放送プラットフォーム”, <https://www.nhk.or.jp/strl/media-platform/>
- [5] IPTVFJ: “HTML5 ブラウザ仕様”, STD-0011, 2.6 版
- [6] “W3C Media Source Extensions”, <https://www.w3.org/TR/media-source-2/>
- [7] “DVB-I”, “<https://dvb-i.tv/>”
- [8] “放送と通信のシームレスな視聴プラットフォーム技術”, <https://www.nhk.or.jp/strl/open2022/tenji/1/index.html>
- [9] “ATSC3.0”, <https://www.atsc.org/atsc-documents/type/3-0-standards/>
- [10] 瀧口, 松村, 藤沢, “端末連携機能の拡張により伝送路抽象化を可能とする受信機アーキテクチャ”, 第20回情報科学技術フォーラム, No.4, M-027, pp.235-236 (2021)
- [11] “The Chromium Projects”, <https://www.chromium.org/>
- [12] “Google Chrome”, <https://www.google.com/intl/ja/chrome/>
- [13] “Microsoft Edge”, <https://www.microsoft.com/ja-jp/edge/>
- [14] “FFmpeg”, <https://ffmpeg.org/>