



---

# GINGA Services Using HD Radio™ Technology

**Revision 1.0**  
**February 27, 2013**

iBiquity Digital Corporation does not make any warranty or representation whatsoever as to the sufficiency or accuracy of information contained in this document. It is further acknowledged that there can be no assurance that this document will not change, perhaps substantially, in the future. Accordingly, iBiquity Digital Corporation assumes no responsibility arising from any present use or misuse of this document.

**Copyright © 2012 – 2013 iBiquity Digital Corporation.**

All rights reserved.

The contents of this publication may not be reproduced in any media without the express written permission of iBiquity Digital Corporation.

**TRADEMARKS**

“iBiquity Digital”, “HD Radio”, and the iBiquity Digital, HD, HD Radio, and “Arc” logos are proprietary trademarks of iBiquity Digital Corporation.

All other trademarks, whether claimed or registered, are the exclusive property of their respective owners.

**iBiquity Digital Corporation**

6711 Columbia Gateway Drive, Suite 500

Columbia, MD 21046

Voice: 443-539-4290

Fax: 443-539-4291

**E-mail address:**

[info@ibiquity.com](mailto:info@ibiquity.com)

# Table of Contents

---

## Contents

<b>1</b>	<b>SCOPE .....</b>	<b>3</b>
1.1	System Overview.....	3
1.2	Document Overview.....	3
<b>2</b>	<b>PROPOSED INTEGRATION OF GINGA USING HD RADIO™ TECHNOLOGY .....</b>	<b>4</b>
2.1	Introduction .....	4
<b>3</b>	<b>OVERVIEW .....</b>	<b>5</b>
3.1	Proposed Software Architecture .....	5
3.2	Station Information Services (SIS).....	5
3.3	Service Information Guide (SIG) .....	5
3.4	Service Information Protocol.....	5
3.5	Structure of Data Objects.....	5
3.6	Request/Response XML API.....	6
3.7	TCP or UDP, UDP Input Port & UDP Output Port, and TCP Listen Port .....	6
<b>4</b>	<b>PROPOSED GINGA RECEIVER IMPLEMENTATION .....</b>	<b>7</b>
4.1	HD Radio Decoder .....	7
4.1.1	Layer 1 Decoder.....	7
4.1.2	SIG Transport.....	7
4.1.3	LOT Transport.....	8
4.1.4	AAS Transport .....	8
4.1.5	Command Processor.....	8
4.2	Host Microprocessor Application.....	8
4.2.1	Radio Application .....	8
4.2.2	Radio Service Processor.....	8
4.2.3	GINGA Application .....	8

## List of Figures

Figure 2-1: Block Diagram of the GINGA System Integration with MSAC.....	4
Figure 4-1 Block diagram of GINGA services on HD Radio receiver .....	7

# 1 Scope

## 1.1 System Overview

The iBiquity Digital Corporation HD Radio™ system is designed to permit a smooth evolution from current analog amplitude modulation (AM) and frequency modulation (FM) radio to a fully digital in-band on-channel (IBOC) system. This system delivers digital audio and data services to mobile, portable, and fixed receivers from terrestrial transmitters in the existing medium frequency (MF) and very high frequency (VHF) radio bands. Broadcasters may continue to transmit analog AM and FM simultaneously with the new, higher-quality, and more robust digital signals, allowing themselves and their listeners to convert from analog to digital radio while maintaining their current frequency allocations.

## 1.2 Document Overview

This document describes proposed middleware that would handle translation of GINGA data services to the Importer multiport configuration. Specifically, it would handle configurations with synchronous and asynchronous image clients attached to GINGA services multiport service.

There exists an implementation of the MSAC in Java, as a server process. The MSAC is a client of the Importer, but as a server component, it has a messaging side that is exposed to its client or clients.

## 2 Proposed Integration of GINGA Using HD Radio™ Technology

### 2.1 Introduction

The HD Radio™ broadcast system employs a system “super” client capable of interfacing with the Importer on one side and with a sequencer of data on the other side via a set of simple XML requests. This client is known as the Multiport Synchronous – Asynchronous Client (MSAC). The proposed implementation calls for the development of a simple translation layer between the DSM-CC Server and the MSAC client.

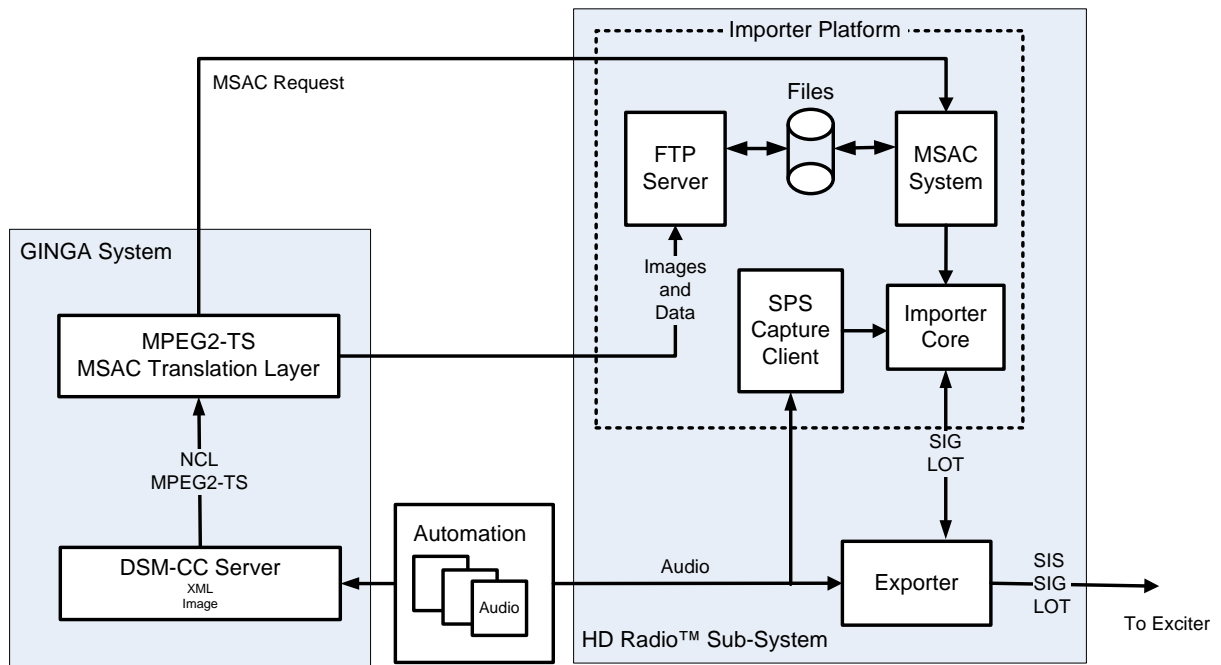


Figure 2-1: Block Diagram of the GINGA System Integration with MSAC

The functionality of the GINGA/MSAC translation layer can be understood by looking at the diagram in Figure 2-1. The MSAC is a Java process with several threads. It communicates to the Importer, via the Importer/Client XML Command/Response API.

A set of configuration files are used to establish the username/password of the multiport client on the Importer, the Importer name or IP address, the directory to read for the files being sent, a file that sets forth the associations between the data clients on the multiport and the available audio clients.

The basic idea is to have a set of configuration files guide the automatic login and open session Command/Responses to the Importer when the MSAC is started. At that point, the Importer will reverse the direction of the Importer/Client XML API and begin to drive Get Data Commands to the client.

## 3 Overview

### 3.1 Proposed Software Architecture

The MSAC is basically a two-thread system. One thread handles all Importer communications and handshaking data through the Importer/Client XML API. The other thread primarily deals with handling the GINGA or other Client/MSAC requests. The Request/Response XML API essentially schedules files to be sent to the Importer. The main work is performed by the *scheduler.getData* call; this method determines which data is going to be sent to the Importer. The Importer will keep asking for data until it has enough to satisfy internal buffer requirements.

### 3.2 Station Information Services (SIS)

The Station Information Service (SIS) provides broadcast station identification and control information. SIS is transmitted in a series of SIS Protocol Data Units (PDUs) on the HD Radio Primary Data Service (PIDS) logical channel. The PIDS channel is a fixed rate channel that delivers basic control messages. Basic control messages would carry GINGA service information. Service information is additional information about the services carried in real time.

### 3.3 Service Information Guide (SIG)

Fundamentally, SIG is a language; a system of terms that includes the rules for combining the terms into structured and meaningful messages that are transmitted to HD Radio receivers for display as announcements to the listener. SIG carries specific information pertinent to receiving data from a service. GINGA will use SIG to identify the service and port assignments on the receiver.

### 3.4 Service Information Protocol

The *service information protocol* is part of the service being broadcast and interacts with the session (or application) layer. The protocol governs the construction of SIG *messages* that are fundamentally structured through the proper use of *information descriptors*. The service information protocol is expandable and new information descriptors may be added as the system develops.

### 3.5 Structure of Data Objects

Requests to send files via the LOT protocol and non-LOT protocol are sent to the MSAC and it queues up the files and determines which packet to send. Each LOT object being sent goes through a series of states which determine when the LOT object will be sent or when packets from it will be sent.

A data object which has been requested will start out in a PENDING state. At this point it will have been copied from the directory specified in the message to a working directory within the relative path of the MSAC distribution.

If it will be sent asynchronously, it will transition to the ACTIVE state on the first chance it gets to run. Once in the ACTIVE state, it is a candidate for having packets sent to the Importer. Objects will remain in the ACTIVE STATE until they are canceled or replaced. In either case, the old file is transitioned to the TERMINATED state and the file is deleted from the working directory.

### 3.6 Request/Response XML API

The Request/Response XML API is typically an API between the MSAC, that is receiving requests, and one or more automation systems or processes, which are requesting files be sent to the Importer.

### 3.7 TCP or UDP, UDP Input Port & UDP Output Port, and TCP Listen Port

The physical implementation of the channel between the studio automation and the GINGA and MSAC process is a UDP datagram or a TCP connection. For UDP, there is one port that the MSAC listens on and one port that it responds to. It will respond to the IP address sending in the Request. There is no authentication on this UDP port exchange; this is assumed to be operating in a friendly system environment. On setup, it is envisioned that the MSAC will run on the Importer and the studio automation equipment will implement the Request/Response MSAC XML API. In fact, several studio automation systems could be talking to one MSAC with this interface. Alternately, the MSAC or a set of MSACs could run on a centralized server or server farm and have the WAN connection between the MSAC and the Importers. The MSAC was implemented as a server process with enough flexibility to accommodate a variety of implementation options.



## 4 Proposed GINGA Receiver Implementation

The HD Radio receiver architecture consists of RF tuner section, HD Radio decoder, and host application microprocessor. The GINGA process will run as an application on the radio host micro and manage the content transfer and content rendering between the HD Radio decoder and the radio display driver.

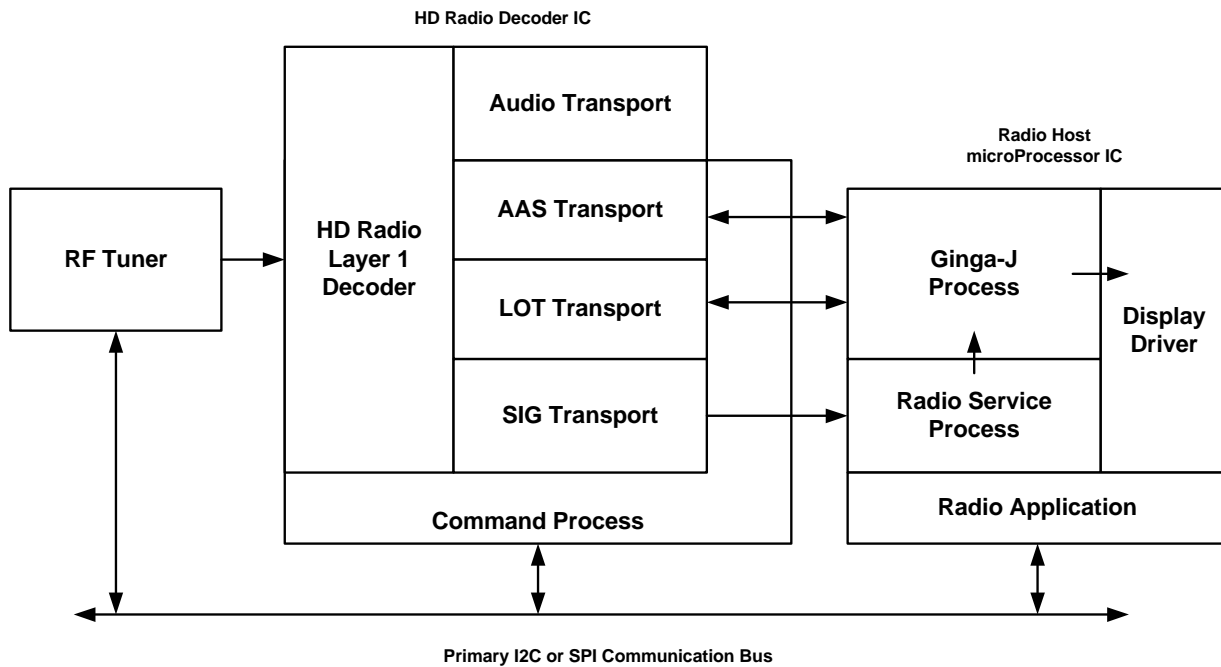


Figure 4-1 Block diagram of GINGA services on HD Radio receiver

### 4.1 HD Radio Decoder

The HD Radio decoder consists of several processing elements to decode the OFDM waveform and extract the services from the broadcast.

#### 4.1.1 Layer 1 Decoder

The Layer 1 decoder provides demodulation function of the OFDM waveform. This process acquires the OFDM waveform, synchronizes the OFDM symbol, and employs error correction techniques to minimize errors in the received message.

The received message is parsed into various transport applications to acquire specific services.

#### 4.1.2 SIG Transport

The SIG transport provides station information identifiers for each audio and data services in the HD Radio broadcast. Accessing SIG allows the receiver to detect which types of services are available and which ports to access these services.

The GINGA service will have a unique identifier in SIG. The host micro application will use SIG to identify the GINGA service and any timing controls for that service.

#### 4.1.3 LOT Transport

The LOT transport provides large object content such as image files, movie files, sound files, or any other service content of significant size. LOT files consist of unique file names and are aggregated in the HD Radio decoder.

The LOT transport will be used for transmission of the content files associated with the GINGA application.

#### 4.1.4 AAS Transport

The AAS transport provides packet and streaming services for small data objects. This transport layer enables smaller services to have rapid acquisition. Each AAS service has a unique identifier in SIG entry.

The AAS transport will be used for the GINGA NCL script files. Each script file is 184 bytes in size and will be easily supported through the packet objects in AAS.

#### 4.1.5 Command Processor

The command processor manages communication interface between the host microprocessor and the HD Radio decoder. Unique commands allow programmers to access the various data tables (SIG, LOT and AAS). All content files must be requested by the host microprocessor.

### 4.2 Host Microprocessor Application

#### 4.2.1 Radio Application

The radio application in the host microprocessor is responsible for managing all the general radio commands. This set includes tuning frequency, control of the HD Radio decoder, and receive status evaluation.

Each HD Radio decoder has a well documented set of commands to enable access to various functions in the decoder IC.

#### 4.2.2 Radio Service Processor

The radio service processor is a function of the radio application. This process is responsible for detecting various radio services in the SIG table. The radio service process would validate the GINGA service for the current radio frequency. Following verification, the radio service process would extract data port IDs and other service acquisition parameters.

#### 4.2.3 GINGA Application

The GINGA application is responsible for decoding the NCL script files (from AAS transport) and calling for the associated content files such as images or audio or text (from the LOT transport). The GINGA application will render the content on the display per the script instructions from the NCL file.