



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

PROYECTO DE GRADO PREVIO A LA OBTENCIÓN DE TÍTULO DE INGENIERO
ELECTRÓNICO EN TELECOMUNICACIONES

AUTOR: MENA BASTIDAS, ROBERTO JAVIER

TEMA: DESARROLLO DE UNA PLATAFORMA DE CONTROL PARA LA
TARJETA MODULADORA DE TELEVISIÓN DIGITAL DEKTEC DTU-215

DIRECTOR: DR. OLMEDO, GONZALO
CODIRECTOR: ING. ACOSTA, FREDDY

SANGOLQUI – ECUADOR

2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE

INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICADO

Dr. Gonzalo Olmedo C.

Ing. Freddy Acosta B.

CERTIFICAN

Que el trabajo titulado “DESARROLLO DE UNA PLATAFORMA DE CONTROL PARA LA TARJETA MODULADORA DE TELEVISIÓN DIGITAL DEKTEC DTU-215”, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el reglamento de estudiantes de la Universidad de las Fuerzas Armadas.

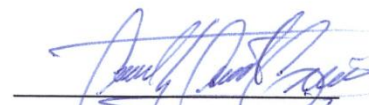
Este trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato de documento portátil (pdf)

Sangolquí, 30 de marzo del 2015.



Dr. Gonzalo Olmedo

DIRECTOR



Ing. Freddy Acosta

CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

ROBERTO JAVIER MENA BASTIDAS

DECLARO QUE:

El proyecto de grado titulado “DESARROLLO DE UNA PLATAFORMA DE CONTROL PARA LA TARJETA MODULADORA DE TELEVISIÓN DIGITAL DEKTEC DTU-215”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas constan en el documento, cuyas fuentes se incorporan en la bibliografía. Consecuente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, 30 de marzo del 2015



Roberto Mena B.

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN

ROBERTO JAVIER MENA BASTIDAS

Autorizo a la Universidad de las Fuerzas Armadas – ESPE, la publicación en la biblioteca virtual de la Institución el trabajo “DESARROLLO DE UNA PLATAFORMA DE CONTROL PARA LA TARJETA MODULADORA DE TELEVISIÓN DIGITAL DEKTEC DTU-215”, cuyo contenido, ideas y criterios es de mi exclusiva responsabilidad y auditoría.

Sangolquí, 30 de marzo del 2015



Roberto Mena B.

DEDICATORIA

A mis sueños.

AGRADECIMIENTO

A Don Paco y Doña Tere por ser el pilar fundamental durante toda mi vida. Agradezco su educación, apoyo incondicional y amor.

A mis hermanos. Un ejemplo de vida que siempre los tengo presente,
a mi manera.

A Señorita Ray, el complemento y fortaleza durante este tiempo.

A mis amigos, profesores, compañeros. A todas las personas con las que he caminado este trayecto, con los que compartí muchas experiencias, con los que me caí y me ayudaron a
levantar.

Gracias!

INDICE GENERAL

INDICE GENERAL	vii
INDICE DE FIGURAS	xi
INDICE DE TABLAS	xiv
GLOSARIO	xvi
RESUMEN	xx
ABSTRACT	xxi
MARCO TEÓRICO	1
1.1 TELEVISIÓN DIGITAL TERRESTRE	1
1.1.1 Introducción.....	1
1.1.2 Estándares de TDT	2
1.1.2.1 Estándar ATSC	3
1.1.2.2 Estándar DVB-T	4
1.1.2.3 Estándar DVB-T2	5
1.1.2.4 Estándar DTMB	5
1.1.2.5 Estándar ISDB-T.....	6
1.1.2.6 Estándar ISDB-Tb.....	7
1.2 ESTÁNDAR ISDB-Tb	7
1.2.1 Estructura General	7
1.2.1.1 Bloque de código fuente	8
1.2.1.2 Bloque de multiplexación	8

1.2.1.3	Bloque de transmisión.....	9
1.2.2	Transmisión Jerárquica.....	9
1.2.3	Transport Stream	11
1.2.3.1	Cabecera Paquete TS	12
1.2.3.2	Broadcast Transport Stream.....	13
1.2.4	Sistema de Transmisión ISDB-Tb.....	13
1.2.4.1	Remultiplexación del Transport Stream	15
1.2.4.2	Codificación Externa.....	15
1.2.4.3	División en Capas Jerárquicas	16
1.2.4.4	Dispersión de Energía	16
1.2.4.5	Ajuste de atraso	17
1.2.4.6	Entrelazador de Byte	17
1.2.4.7	Codificación Interna.....	18
1.2.4.8	Entrelazador de bit	19
1.2.4.9	Modulación de la portadora	19
1.2.4.10	Combinación de capas jerárquicas	21
1.2.4.11	Entrelazador de tiempo	22
1.2.4.12	Entrelazador de frecuencia.....	23
1.2.4.13	Señal OFDM	23
1.2.5	Capacidad de Transmisión del Sistema	25
1.2.6	Tarjetas Moduladoras ISDB-Tb	26
1.2.6.1	Kathrein NDS-3542 Encoder Modulator ISDB-T	26
1.2.6.2	PROMAX MO-370 LE.....	27
1.2.6.3	TECSYS TS 9500-ISDB-T.....	28
1.2.6.4	DekTec DTA-115	28
	PLATAFORMA DE TRANSMISIÓN DEKTEC	29

2.1	Tarjeta Moduladora DekTec DTU-215	29
2.1.1	Características Generales.....	30
2.2	Software DekTec Streamxpress.....	31
2.2.1	Características Generales.....	31
2.2.2	Transmisión de TS con el estándar ISDB-Tb.....	34
2.2.2.1	Mapeamiento del TS en las capas jerárquicas	34
2.2.2.2	Parámetros ISDB-Tb.....	35
2.2.2.3	Parámetros de la Capa Jerárquica	36
2.2.2.4	Centro de Mensajes	37
2.2.3	Requerimientos de Hardware	37
2.3	INTERFAZ DE PROGRAMACIÓN DE APLICACIONES DekTec	38
2.3.1	Descripción General	38
2.3.2	Contenido de Clases ISDB-Tb	39
2.3.2.1	Clase DtDevice	40
2.3.2.2	Clase DtIsdbtPars	41
2.3.2.3	Clase DtOutpChannel	43
2.3.3	Contenido de Estructuras ISDB-Tb.....	48
2.3.3.1	Estructura DtDeviceDesc	48
2.3.3.2	Estructura DtIsdbtLayerPars	49
	DESARROLLO DEL SOFTWARE	51
3.1	Entornos de desarrollo integrado	52
3.1.1	IDE Visual Studio 10.0.....	52
3.1.1.1	Características Generales	52
3.1.2	IDE Netbeans.....	53
3.1.2.1	Características Generales	54
3.2	Desarrollo Aplicación	54
3.2.1	Código Java	55

3.2.1.1	Clase IntegracionJNI.....	55
3.2.1.2	Clase ReproducciónVideo.....	56
3.2.1.3	Clase LecturaPids.....	57
3.2.1.4	Clase GUI_Main	57
3.2.2	Integración Java/C++.....	60
3.2.2.1	Implementación JNI.....	61
3.2.3	Código C++	64
PRUEBAS Y ANALISIS DE RESULTADOS		75
4.1	Pruebas y análisis de resultados para archivos broadcast transport stream	76
4.2	Pruebas y análisis de resultados para archivos transport stream	81
4.3	Comparación entre la plataforma de control desarrollada y stream Xpress	83
4.3.1	Interfaz Gráfica de Usuario	83
4.3.2	Transmisión de Archivos Transport Stream.....	84
4.4	PRUEBAS EN OTRA PC	84
CONCLUSIONES Y RECOMENDACIONES		86
5.1	CONCLUSIONES	86
5.2	RECOMENDACIONES.....	87
BIBLIOGRAFIA		89

INDICE DE FIGURAS

Figura 1.1 Adopción estándares TDT nivel mundial	2
Figura 1.2 Estructura de transmisor del estándar ATSC	4
Figura 1.3 Estructura de transmisor del estándar DVB-T	4
Figura 1.4 Estructura de transmisor del estándar ISDB-T.....	6
Figura 1.5 Estructura del Estándar ISDB-Tb.....	8
Figura 1.6 Generación Transport Stream	9
Figura 1.7 Orden de los 13 segmentos en canal de 6 MHz	10
Figura 1.8 Ejemplo distribución de los segmentos en capas	10
Figura 1.9 Cabecera paquete TS.....	12
Figura 1.10 Diagrama de bloques del sistema de transmisión del estándar ISDB-Tb	14
Figura 1.11 Transport Stream re multiplexado (modo 1, intervalo de guarda 1/8).....	15
Figura 1.12 Paquete TS sin codificación RS(a) y con codificación RS(b).....	16
Figura 1.13 Generador de secuencias pseudo aleatorias	17
Figura 1.14 Circuito de byte interleaving.....	17
Figura 1.15 Circuito de codificación de código convolucional.....	18
Figura 1.16 Estructura de los entrelazadores de bits	19
Figura 1.17 Constelación QPSK.....	19
Figura 1.18 Constelación DQPSK.....	20
Figura 1.19 Constelación 16QAM.....	20
Figura 1.20 Constelación 64QAM.....	21
Figura 1.21 Combinador capas jerárquicas	21
Figura 1.22 Configuración de la sección de entrelazado de tiempo.....	22
Figura 1.23 Ejemplo entrelazamiento de tiempo sobre un ruido impulsivo.....	22
Figura 1.24 Efecto del entrelazamiento de frecuencia	23

Figura 1.25 División ortogonal multiplexada.....	23
Figura 1.26 Señal OFDM	24
Figura 1.27 Kathrein NDS-3542	27
Figura 1.28 Aplicaciones típicas Kathrein NDS-3542	27
Figura 1.29 PROMAX MO-370 LE	27
Figura 1.30 TECSYS TS 9500-ISDB-T	28
Figura 1.31 DekTec DTA-115	28
Figura 2.1 DekTec DTU-215.....	31
Figura 2.2 GUI StreamXpress	32
Figura 2.3 Áreas de información del TS.....	34
Figura 2.4 Sección de mapeamiento de capas jerárquicas.....	35
Figura 2.5 Sección de parámetros ISDB-Tb.....	36
Figura 2.6 Sección de parámetros de capa jerárquica	37
Figura 2.7 Sección Centro de mensajes.....	37
Figura 2.8 Ejemplo de Objetos DTAPI sobre moduladora DTU-215.....	39
Figura 3.1 Niveles de Abstracción de la Plataforma de Control para la DTU-215.....	51
Figura 3.2 Componentes principales Visual Studio 2010 [16].....	53
Figura 3.3 Interfaz gráfica de Visual Studio 201.....	53
Figura 3.4 Interfaz gráfica de Netbeans.....	54
Figura 3.5 Diagrama UML Aplicación Java	55
Figura 3.6 Interfaz gráfica de la plataforma de control DTU-215.....	57
Figura 3.7 Integración Java y C++ con JNI.....	60
Figura 3.8 Declaración de métodos nativos en Java.....	61
Figura 3.9 Generación de cabecera utilizando <i>javah</i>	61
Figura 3.10 Utilización de InegracionJNI.h para implementar los métodos nativos	62
Figura 3.11 Configuración proyecto C++ como Librería Dinámica	62
Figura 3.12 Inclusión de directorios adicionales en proyecto C++	63
Figura 3.13 Compilación proyecto C++	63
Figura 3.14 Inclusión directorio ubicación Librería Dinámica.....	64
Figura 4.1 Escenario de Pruebas	75
Figura 4.2 Escenario de pruebas de transmisión y recepción de la señal ISDB-Tb.....	76

Figura 4.3 Configuración utilizada para primera prueba de transmisión	77
Figura 4.4 Captura de la TV durante la recepción de la primera señal de prueba.....	78
Figura 4.5 Analizador de Televisión Agilent CXA.....	78
Figura 4.6 Verificación Frecuencia y Ancho de Banda de Transmisión.....	79
Figura 4.7 Captura parámetros de Tx capa jerárquica A.....	79
Figura 4.8 Captura parámetros de Tx capa jerárquica B	80
Figura 4.9 Decodificación Información TMCC	81
Figura 4.10 Configuración utilizada para segunda prueba de transmisión.....	81
Figura 4.11 Captura de la TV durante la recepción de la segunda señal de prueba	82
Figura 4.12 Captura parámetros de Tx capa jerárquica A.....	82
Figura 4.13 Comparación GUI Presente Proyecto y Stream Xpress	83
Figura 4.14 Frecuencia y Ancho de Banda de Transmisión utilizando Stream Xpress	84

INDICE DE TABLAS

Tabla 1.1 Configuración recomendada Capas de Transmisión	11
Tabla 1.2 Configuración de la multiplexación del cuadro multiplex	15
Tabla 1.3 Tasa del código interno y secuencia de la señal de transmisión.....	18
Tabla 1.4 Duración de símbolo y cuadro OFDM	25
Tabla 2.1 Atributos Técnicos DTU-215	30
Tabla 2.2 Detalle componentes generales de la interfaz gráfica de StreamXpress	32
Tabla 2.3 Detalle de la sección de mapeamiento de capas jerárquicas	34
Tabla 2.4 Detalle de la sección parámetros ISDB-Tb	35
Tabla 2.5 Detalle de la sección parámetros de capa jerárquica	36
Tabla 2.6 Detalle de la sección Centro de mensajes.....	37
Tabla 2.7 DtDevice::AttachToType	40
Tabla 2.9 DtDevice::VpdRead	40
Tabla 2.8 DtDevice::Detach	41
Tabla 2.10 class DtIsdbtPars	41
Tabla 2.12 DtIsdbtPars::ComputeRates	42
Tabla 2.11 DtIsdbtPars::CheckValidity.....	43
Tabla 2.13 DtIsdbtPars::RetrieveParsFromTs	43
Tabla 2.14 DtOutpChannel::AttachToPort.....	44
Tabla 2.15 DtOutpChannel::ClearFifo	44
Tabla 2.16 DtOutpChannel::ClearFlags	44
Tabla 2.17 DtOutpChannel::Detach	45
Tabla 2.18 DtOutpChannel::GetFlags	45
Tabla 2.19 DtOutpChannel::SetFifoSize.....	45
Tabla 2.20 DtOutpChannel::SetModControl.....	46

Tabla 2.21 DtOutpChannel::SetTsRateBps	47
Tabla 2.23 DtOutpChannel::SetTxMode.....	47
Tabla 2.22 DtOutpChannel::SetTxControl.....	48
Tabla 2.24 Struct DtDeviceDesc	49
Tabla 2.25 Struct DtIsdbtLayerPars	49
Tabla 3.1 Métodos declarados en la Clase IntegracionJNI	56
Tabla 3.2 Detalle de los elementos de la interfaz gráfica de la aplicación en Java.....	58
Tabla 3.3 Método LeerError	65
Tabla 3.4 Método AdjuntarTarjeta	66
Tabla 3.5 Método ConectarPuerto.....	66
Tabla 3.6 Método LimpiarBanderas	66
Tabla 3.8 Detalle cadena de variable <i>int</i> para configurar parámetros de transmisión.....	67
Tabla 3.9 Método ConfiguraciónTxCapaA	69
Tabla 3.7 Método AbrirArchivo	69
Tabla 3.10 Método ConfiguracionTx	70
Tabla 3.11 Método InitTxParams	71
Tabla 3.12 Método ReproducirTx	72
Tabla 3.13 Método PararTs	73
Tabla 3.14 Método LeerBanderas	73
Tabla 3.15 Método CerrarPrograma.....	73
Tabla 3.16 Método ValidarArchivoTS	74
Tabla 4.1 Configuración ISBD-T para pruebas de laboratorio	76

GLOSARIO

A

- ABNT** Associação Brasileira de Normas Técnicas.
- ARIB** (*Association of Radio Industrie and Businesses*). Asociación de Radio Industrias y Negocios.
- ATSC** (*Advanced Television Systems Committee*). Comité de Sistemas de Televisión Avanzados.

B

- BTS** (*Broadcast Transport Stream*). Flujo de Transporte de Radiodifusión

D

- DLL** (*Dynamic Link Library*). Librería de Enlace Dinámico.
- DQPSK** (*Diferencial Quadrature Phase Shift Keying*). Modulación de Fase en Cuadratura Diferencial.
- DTAPI** (*Dek Tec Application Programming Interface*). Interfaz de Programación de Aplicaciones DekTec.
- DTMB** (*Digital Terrestrial Multimedia Broadcasting*). Radiodifusión Digital Terrestre de Multimedia.

-
- DVB-T** (*Digital Video Broadcasting Terrestrial*). Radiodifusión de Televisión Digital Terrestre.
- DVB-T2** (*Digital Video Broadcasting Terrestrial 2nd Generation*). Radiodifusión de Televisión Digital Terrestre 2da Generación.
- F**
- FIFO** (*First In First Out*) Primero en Entrar, Primero en Salir.
- G**
- GINGA** Middleware Abierto del Sistema Brasileño de Televisión Digital.
- H**
- HD** (*High Definition*). Definición Alta.
- I**
- IDE** (*Integrated Development Environment*). Entorno de Desarrollo Integrado.
- ISDB-T** (*Integrated Services Digital Broadcasting Terrestrial*). Servicios Integrados de Radiodifusión Digital Terrestre.
- ISDB-Tb** (*Integrated Services Digital Broadcasting Terrestrial Brazil*). Servicios Integrados de Radiodifusión Digital Terrestre Brasil.
- J**
- JDK** (*Java Development Kit*). Kit de Desarrollo de Java.
- JNI** (Java Native Interface). Interface Nativa de Java.
- JRE** (*Java Runtime Environment*). Entorno de Ejecución de Java.

M

MPEG-2 (Moving Picture Experts Group 2) Grupo de Expertos de Imágenes en Movimiento Versión 2

O

OFDM (Orthogonal Frequency Division Multiplexing). Multiplexación por División Ortogonal de Frecuencias.

One-Seg (One Segment). Servicio de transmisión digital terrestre y datos complementarios para dispositivos móviles.

P

PES (*Packet Elementary Stream*). Flujo Elemental Paquetizado.

PID (*Packet Identifier*). Identificador de Paquetes.

Q

QAM (*Quadrature Amplitude Modulation*). Modulación de Amplitud en Cuadratura.

QPSK (Quadrature Phase Shift Keying). Modulación de Fase en Cuadratura.

S

SBTVD Sistema Brasileño para Televisión Digital.

SD (*Standard Definition*). Definición Estándar.

T

TD-COFDM (*Time Domain Coded Orthogonal Frequency Division Multiplex*).

Multiplexación por División Ortogonal de Frecuencias Codificadas en Dominio del Tiempo.

TDT Televisión Digital Terrestre.

TMCC (*Transmission and Multiplexing Configuration Control*). Configuración de Control de Transmisión y Multiplexación.

TS (*Transport Stream*). Flujo de Transporte.

U

UHF (*Ultra High Frequency*). Frecuencias Ultra Altas.

UIT Unión Internacional de Telecomunicaciones.

V

VSB (*Vestigial Side Band*). Modulación de Banda Lateral Vestigial.

RESUMEN

El presente proyecto contempló el desarrollo de un software de control que soporte el estándar ISDB-Tb para la tarjeta moduladora DekTec DTU-215; permitiendo a los usuarios del Laboratorio de Televisión Digital manipular y controlar los parámetros de transmisión de la tarjeta moduladora para, como resultado final, lograr enviar por un canal de RF los Transport Stream y Broadcast Transport Stream generados previamente. Es importante recalcar que al analizar detalladamente los parámetros de configuración y bloques de procesos se pudo aprovechar las funcionalidades del equipo. En la primera etapa del proyecto se estudió y analizó el estándar ISDB-Tb, así como las características y funcionalidades de la tarjeta DekTec DTU-215. En la segunda etapa del proyecto se desarrolló una librería dinámica escrita en C++ y que permite utilizar las funcionalidades más importantes de la moduladora. Durante la última etapa se utilizó la librería previamente mencionada y se acopló a una aplicación en java que contiene una interfaz gráfica de usuario amigable y que permite implementar nuevas funcionalidades a futuro. Además se validó el funcionamiento de la plataforma desarrollada utilizando Televisores que reciben el estándar ISDB-Tb y el analizador de televisión.

PALABRAS CLAVES:

- ISDB-TB
- MODULADORA
- TRANSMISIÓN
- TRANSPORT STREAM
- LIBRERÍA DINÁMICA

ABSTRACT

This project was about the development of software that controls the DekTec DTU-215 modulator. This software supports the ISDB-Tb standard and permits the users of the Digital Television Laboratory manipulate and control the transmission parameters of this DekTec device. As result, users can transmit through an RF channel video and audio enclosed in a Transport Stream file. It is important to mention that analyze the configuration parameters and process blocks allows to use all the functionalities of the modulator. On the first stage of this project, Besides the DTU-215's features and functionalities studied, the ISDB-Tb standard was closely analyzed. This study was emphasized in the physical link of the standard. The second stage was about the development of a dynamic link library written in C++ code that can control the modulator DTU-215. In addition, the previously mentioned library was used to elaborate a graphic application in Java that has more functionalities which can be improved in the future. Tests were made using Digital Televisions and the TV Analyzer CXA from Agilent.

KEY WORDS:

- ISDB-TB
- MODULATOR
- TRANSMISSION
- TRANSPORT STREAM
- DYNAMIC-LINK LIBRARY

CAPÍTULO I

1. MARCO TEÓRICO

1.1 TELEVISIÓN DIGITAL TERRESTRE

1.1.1 Introducción

La televisión es el principal medio de comunicación masivo en el Ecuador. Este ha estado presente en los hogares de los ecuatorianos desde diciembre de 1960 [1] y desde entonces se ha podido contemplar su evolución a las necesidades emergentes del usuario.

La Televisión Digital Terrestre (TDT) es el conjunto de soluciones tecnológicas que permiten la transmisión y recepción de imágenes en movimiento, sonido asociado y datos a través de señales digitales sobre una red de repetidoras terrestres.

La transmisión de señales digitales en televisión permite aportar diversas ventajas sobre la televisión analógica. Se puede mencionar que las principales son la mejora de la calidad de imagen y sonido, la posibilidad de comprimir la señal y efectuar un uso más eficiente del espectro radioeléctrico, además de aumentar el número de programas transmitidos

Es necesario recalcar que el aprovechamiento del ancho de banda permite implementar servicios como guía electrónica de programación, servicios interactivos, acceso a internet, etc. La interactividad de la televisión digital permite participar en encuestas, concursos, visualizar información del tráfico y otras aplicaciones que generan un experiencia distinta al telespectador [2].

1.1.2 Estándares de TDT

Distintos estándares de Televisión Digital Terrestre se han desarrollado alrededor del mundo. Estos han sido aprobados y adoptados en diferentes países considerando los servicios y características técnicas que ofrecen. Actualmente existen cuatro estándares, además de dos variantes relacionadas a dos de ellos.

- Advanced Television Systems Committee (ATSC), Estándar Americano
- Digital Video Broadcasting – Terrestrial (DVB-T), Estándar Europeo
- Digital Video Broadcasting – Terrestrial2 (DVB-T2) Estándar Europeo 2da Generación
- Digital Terrestrial / Television Multimedia Broadcasting (DTMB), Estándar Chino
- Integrated Services Digital Broadcasting – Terrestrial (ISDB-T), Estándar Japonés
- Integrated Services Digital Broadcasting – Terrestrial Brazilian version (ISDB-Tb), Estándar Japonés-Brasileño

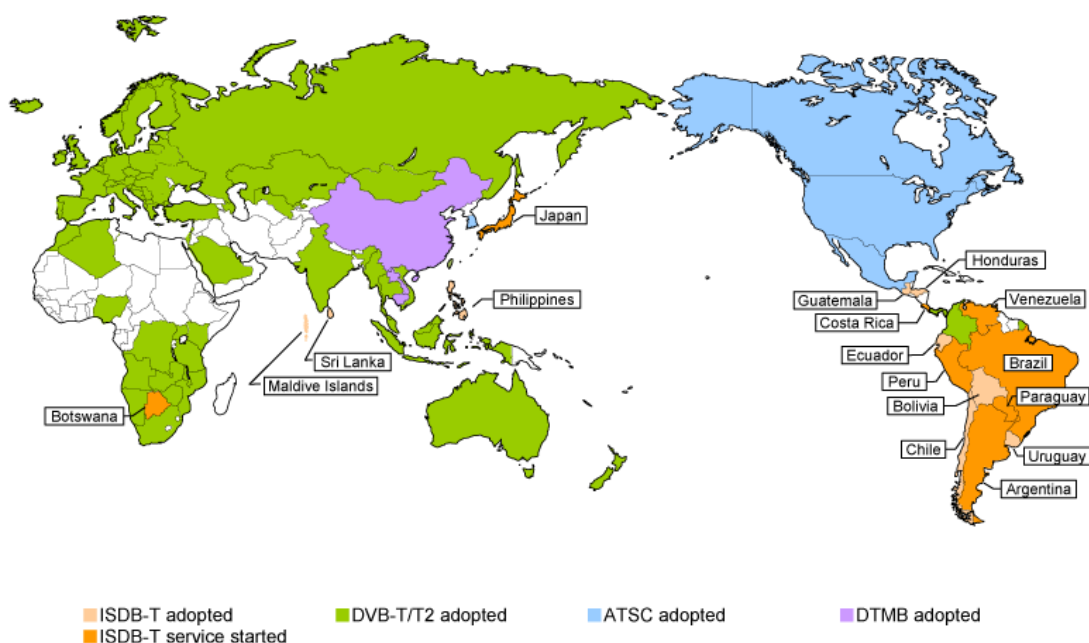


Figura 1.1 Adopción estándares TDT nivel mundial

En la figura 1.1 se observa los estándares de TDT adoptados en el mundo (actualizado agosto del 2014) [3], tomando en cuenta que los países que aún no han adoptado sistema alguno, se encuentran realizando estudios comparativos de los distintos estándares para satisfacer sus necesidades.

1.1.2.1 Estándar ATSC

El estándar de televisión digital ATSC describe un sistema diseñado para transmitir video de alta calidad, audio y datos auxiliar sobre un canal de 6 MHz. El sistema puede transmitir confiablemente tasas cerca de 19 Mbps. ATSC es el estándar norteamericano, el mismo que fue adoptado en varios países como Canadá, Estados Unidos, México, etc. ATSC emplea una modulación 8-VSB (Vestigial Side Band) con una codificación de Trellis que permite veracidad y velocidad en las transmisiones a realizar. Se emplea MPEG-2 para la compresión de audio y video.

El objetivo es maximizar la información que pasa a través del canal de datos, reduciendo al mínimo la cantidad de datos requeridos para representar la secuencia de imágenes de vídeo y su audio asociado. Se representa el video, el audio y las fuentes de datos con el menor número de bits como sea posible mientras se preserva el nivel de calidad requerido para la aplicación dada.

El sistema ATSC está definido por tres subsistemas:

- Codificación y compresión de los datos fuente
- Multiplexación y transporte del servicio
- Transmisión TF [4]

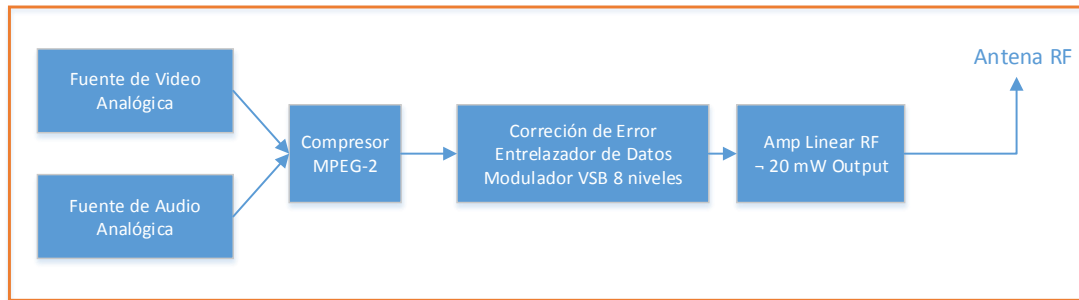


Figura 1.2 Estructura de transmisor del estándar ATSC

1.1.2.2 Estándar DVB-T

Es el estándar para la transmisión de TDT creado por la organización europea Digital Video Broadcasting (DVB). DVB-T describe el sistema que transmite video, audio y otros datos a través de un flujo MPEG-2 usando una codificación COFDM (Coded Orthogonal Frequency Division Multiplexing).

El presente estándar adecua la señal al medio de difusión, elige el método de modulación, define el código común de corrección de errores y codificación dependiendo del canal de comunicación, además marca el método de cifrado y también fija la interfaz común de acceso condicional. DVB-T es usado fundamentalmente en la transmisión de múltiples programas en un solo canal.

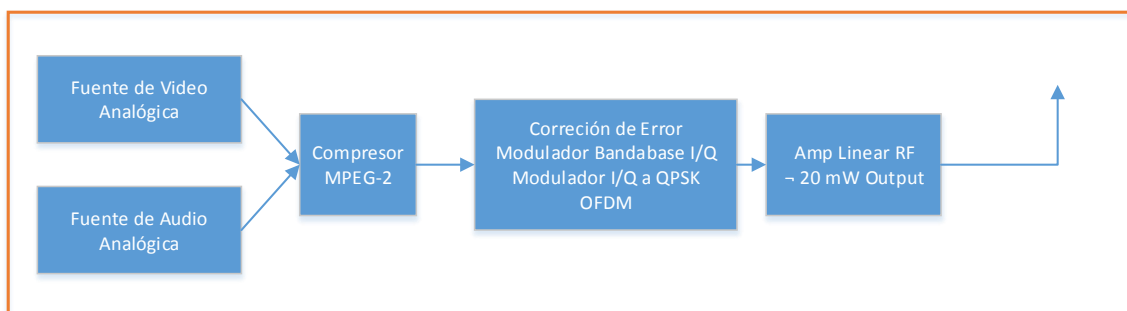


Figura 1.3 Estructura de transmisor del estándar DVB-T

1.1.2.3 Estándar DVB-T2

Similar a su predecesor, DVB-T2 usa modulación OFDM con un gran número de portadoras que transmiten una señal más robusta. Por otro lado, su rango variado de modos permite ser un estándar flexible. El número de portadoras, intervalo de guarda y señales pilotos pueden ser manipulados permitiendo optimizar el uso de espectro en la transmisión. Las nuevas características usadas en DVB-T2 se detallan a continuación:

- Codificación Alamouti que es un método de diversificación de transmisores que mejora la cobertura en redes de frecuencia simple y escala pequeña.
- Rotación de Constelación provee robustez adicional para constelaciones de orden inferior.-
- Interleaving Extendido incluye interleaving de bit, celda, tiempo y frecuencia.
- Future Extension Frames (FEF) permite al estándar una evolución compatible en un futuro.

Como resultado, DVB-T2 ofrece una mayor tasa de transmisión y mayor robustez que su predecesor. DVB-T tiene una tasa máxima de transmisión de 23,75 Mbps y DVB-T2 de 40 Mbps [5]

1.1.2.4 Estándar DTMB

Desarrollado en la Universidad de Tsinghua en la República Popular de China y definida en el 2006, presenta características diferentes a otros estándares en el sistema de modulación y codificación de canal. Además, ofrece alta definición de la imagen, movilidad, etc.

Es un estándar únicamente utilizado en China, Hong Kong y Macao, utiliza TD-COFDM (Time Domain Coded Orthogonal Frequency Division Multiplex) como método de modulación. Mantiene las siguientes características

- Codificación de canal.
- Modulación QAM digital.
- Codificación LPDC FEC.
- Soporta tasas de transmisión desde 4.813 Mbps hasta 32.486 Mbps.
- Secuencia PN para el encabezado de la trama
- MFN y SFN

1.1.2.5 Estándar ISDB-T

Fue desarrollado por ARIB (Association of Radio Industrie and Businesses) en Japón años después que DVB-T y ASTC, por lo que tuvo en cuenta la experiencia de los otros estándares para mejorar su rendimiento. Usa el sistema de multi-portadoras COFDM. Sus canales de 6 MHz pueden ser divididos en 13 sub-bandas, donde varios parámetros de modulación pueden ser seleccionados.

El estándar utiliza sistema compresor de video MPEG-2 y de audio MPEG-L2. Es importante mencionar que ISDB-T permite transmisión fija y móvil, también conocida como one-seg. Puede utilizar modulaciones QPSK, DQPSK, 16QAM y 64QAM además de otras prestaciones que lo convierte en un sistema flexible y robusto, a prueba de errores.

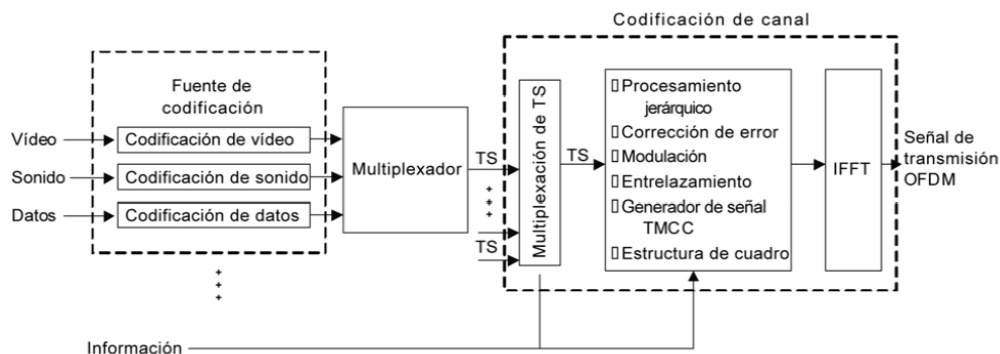


Figura 1.4 Estructura de transmisor del estándar ISDB-T

1.1.2.6 Estándar ISDB-Tb

La versión Brasileña del estándar Japonés usa codificación MPEG-4 en video y HE-AAC en el audio, también introduce el middleware Ginga, desarrollado en Brasil, que permite la interactividad por parte del usuario.

En este estándar, la capa física mantiene las mismas características que su versión predecesora y ha sido adoptado por la mayoría de países Latinoamericanos.

El 29 de abril del 2009 la UIT (Unión Internacional de Telecomunicaciones) certifico el estándar ISDB-Tb así como al módulo Ginga-NCL como primera recomendación para entornos multimedia interactivos de televisión digital. En la sección 1.2 se estudia a profundidad el estándar.

1.2 ESTÁNDAR ISDB-TB

En un principio denominado SBTVD (Sistema Brasileño para Televisión Digital) es un sistema basado en el estándar ISDB-T Japonés. Fue desarrollado por un grupo de estudio, coordinado por el Ministerio de Comunicaciones en Brasil y dirigido por la Agencia de Telecomunicaciones Brasileña (ANATEL).

El servicio de One-Segment dentro del mismo ancho de banda, alta eficiencia en la recepción móvil/portátil, robustez contra efecto multi trayectos y uso para casos de prevención de desastres permite al estándar ISDB-Tb ser un sistema de transmisión de televisión con mejores prestaciones técnicas que ATSC y DVB-T.

1.2.1 Estructura General

El estándar ISDB-Tb está compuesto por tres bloques funcionales:

- Bloque de código fuente
- Bloque de multiplexación
- Bloque de transmisión

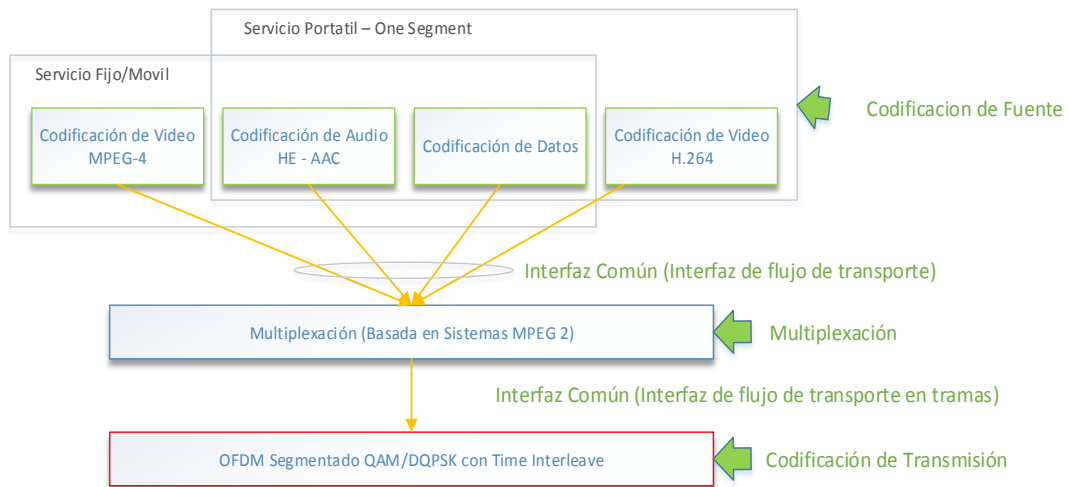


Figura 1.5 Estructura del Estándar ISDB-Tb

1.2.1.1 Bloque de código fuente

El estándar utiliza la compresión de video MPEG-4 para servicios de recepción fija, y compresión de video H.264 para servicio one-segment. La compresión del audio, sea fijo o móvil, está dado por el método HE-ACC. A diferencia de su predecesor Japonés que utilizaba MPEG-2 y MPEG L2 respectivamente, permitiendo una mayor compresión y optimización de recursos.

La codificación de datos está controlada por la plataforma de desarrollo y presentación de contenidos interactivos Ginga, la cual es una capa de software intermedio (middleware) entre el hardware/Sistema Operativo y las aplicaciones interactivas.

1.2.1.2 Bloque de multiplexación

En la parte de multiplexación, igual al estándar Japonés, los contenidos de datos, audio y video son multiplexados en un paquete de flujo de transporte llamado *Transport Stream* (TS). A pesar que cualquier contenido/servicio puede ser multiplexado, el contenido de audio y video pasa por una etapa previa donde son convertidos al formato *Packet Elementary*

Stream (PES) y finalmente al TS. Sin embargo los contenidos que no son tipo de flujo de datos, son convertidos al formato de sección para posteriormente ser transformados en TS. [6]

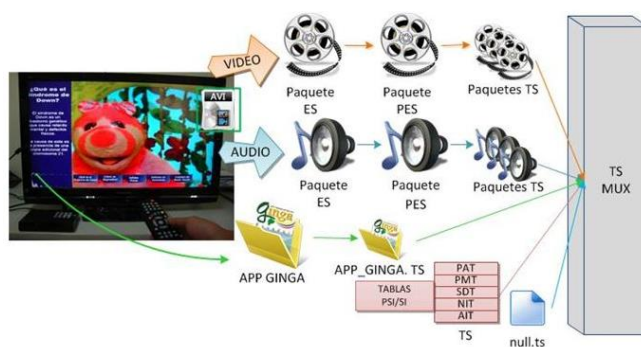


Figura 1.6 Generación Transport Stream

1.2.1.3 Bloque de transmisión

Respecto a la transmisión, el estándar ISDB-Tb mantiene los procesos y las características técnicas que su predecesor japonés El estudio de este bloque se estudia a profundidad en la sección 1.2.3.

1.2.2 Transmisión Jerárquica

La transmisión jerárquica es una característica del sistema ISDB-T, la cual asigna los parámetros de transmisión a cada ID del servicio. Por ejemplo, al servicio que necesita mayor robustez contra ruido urbano y que nos permita la recepción móvil se le asigna modulación QPSK, mientras que a otros servicios se le puede asignar modulación 64QAM.

El estándar ISDB-Tb transmite hasta tres capas independientes (Capas A, B y C) empleando la segmentación de banda. Esta estructura divide al canal de 6 MHz en 14 segmentos OFDM iguales (cada uno 429 kHz) de los cuales se usan 13 en las capas jerárquicas. El último

segmento es dividido en 2 y es utilizado como banda de guarda para evitar interferencia de canales adyacentes.

Es importante mencionar que los 13 segmentos no están ordenados de manera ascendente de izquierda a derecha. Por el contrario, parte del segmento central 0 y luego a la izquierda y derecha de manera intercalada como muestra la figura 1.7.

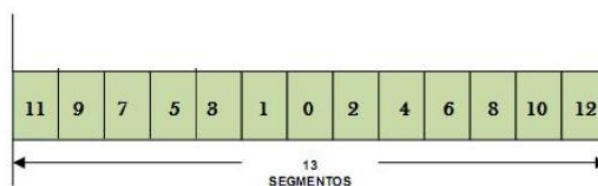


Figura 1.7 Orden de los 13 segmentos en canal de 6 MHz

Aunque no especifica el estándar, el segmento central es usualmente utilizado para recepción parcial y se lo ubica en la capa A. One-seg está destinado para equipos móviles y se maneja en el segmento central ya que los receptores, al sintonizar el canal parten de la frecuencia central, permitiendo así optimizar recursos limitados como es la batería.

Por otro lado, los símbolos transportados por cada capa pueden ser utilizando modulación diferencial (DQPSK) o coherente (QPSK, 16QAM, 64QAM). En caso que alguna capa utilice modulación diferencial, esta debe estar ubicada antes de la capa de modulación coherente.

Por ejemplo, puede existir la siguiente configuración: La capa A es designada para one-seg, capa B para servicios de definición estándar (modulación DQPSK), capa C servicios de alta definición (modulación 64QAM). [7]



Figura 1.8 Ejemplo distribución de los segmentos en capas

Un ejemplo de configuración de las capas para la transmisión se muestra en la tabla 1.1. Donde se observa la transmisión de 3 servicios o programas distintos a definición one-seg, definición estándar y alta definición.

Tabla 1.1 Configuración recomendada Capas de Transmisión

Capa	Número de Segmentos	Tasa de codificación	Modo de Operación	Modulación	Tasa de Video Transmitida
A	1	2/3	Modo 1	QPSK	one-seg 400 kbps
B	8	3/4 o 5/6	Modo 3	64QAM	HD 15 Mbps
C	4	3/4 o 5/6	Modo 2	16QAM	SD 6 Mbps

1.2.3 Transport Stream

Transport Stream es el protocolo de comunicación para video, audio y datos que está definido en los sistemas que trabajan con el estándar MPEG-2 ISO/IEC 13818-1. [8]

El TS es un flujo de la capa de transporte y está conformado por el conjunto de paquetes TS, Estos paquetes TS son de 188 bytes y al ser de tamaño fijo permite mayor facilidad en la detección de errores.

Los paquetes TS permiten multiplexar diferente tipos de datos; los cuales en sus contenidos presentan las características del estándar y la televisión digital. Entre estos servicios podemos mencionar:

- Transmisión de canales One-Seg
- Transmisión de canales *Standard Definition* (SD)
- Transmisión de canales *High Definition* (HD)
- Servidor *Electronic Program Guide* (EPG)
- Servidor de Datos para Interactividad

1.2.3.1 Cabecera Paquete TS

La cabecera de un paquete TS es de 4 bytes y sus 8 campos, como muestra la figura 1.9, cumplen con una función determinada. A continuación se explica brevemente cada uno de estos campos.

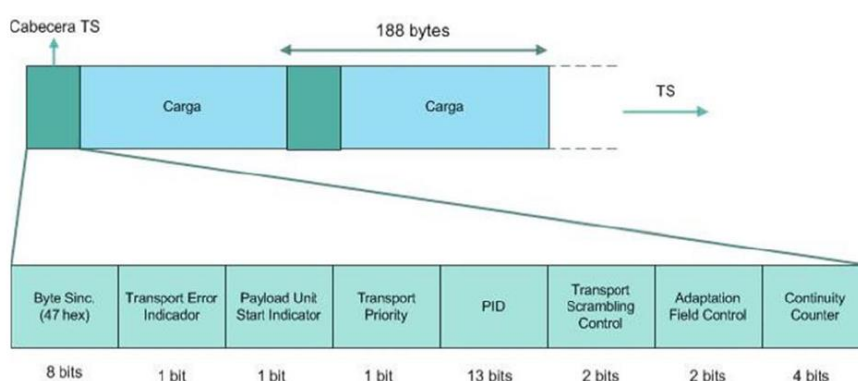


Figura 1.9 Cabecera paquete TS

- *Byte de Sincronismo*: Es utilizado como referencia del inicio del paquete. Tiene un valor fijo de 0x47h (0100 0111).
- *Transport error indicator*: una bandera que indica la presencia de cualquier error de bit en el paquete TS. Al tener el valor '1' indica que el paquete TS tiene un error incorregible de por lo menos un bit.
- *Payload unit start indicator*: cuando contiene el valor '1' indica que el payload de este paquete TS debe empezar en el inicio o en el indicador del paquete PES.
- *PID- Packet identifier*: Formado por 13 bits que identifican cada paquete del TS. El valor del PID depende del tipo de información que está transportando en su carga.
- *Transport priority*: Cuando tiene valor de '1' indica que el paquete transportado tiene mayor prioridad que los demás paquetes con el mismo PID.
- *Transport scrambling control*: Indica si el contenido del paquete sufrió algún tipo de criptografía.

- Adaptation field control: Está formado por 2 bits que indican si entre la cabecera inicial y la carga del paquete existe un campo especial llamado adaptation field.
- Continuity counter: 4 bits que transportan el valor de un contador que es incrementado por cada paquete enviado con el mismo PID.

1.2.3.2 Broadcast Transport Stream

En el TS existe la posibilidad de enviar hacia el sistema de transmisión información de control de transmisión. Por ejemplo la configuración de los segmentos del canal, modulación, intervalo de guarda, etc. Este flujo de transporte ya no se llama TS sino se lo conoce como *Broadcast Transport Stream* (BTS).

En un BTS sus paquetes son de 204 bytes, distinto a los 188 del TS. En estos 16 bytes extras se incluye información ISDBT-Tb, de los cuales 8 bytes indican información sobre la capa, el contador TSP, encabezamientos, datos auxiliares y demás. Los otros 8 bytes restantes son de paridad.

1.2.4 Sistema de Transmisión ISDB-Tb

El Sistema de transmisión del estándar ISDB-Tb o su capa física especifica el de sistema de codificación de canal y modulación, describiendo el procesamiento de señal en el modulador. A continuación se resumen algunos beneficios de este bloque del sistema:

- Eficiencia en el ancho de banda utilizado

La transmisión jerárquica son servicios para diferente tipo de recepción en un solo canal dentro del mismo ancho de banda. La adopción de esta en conjunto con la integración de OFDM permite optimizar recursos en la utilización de la frecuencia asignada para su transmisión.

- Servicio Móvil/Handheld en transmisión estándar

El sistema de transmisión permite entrega servicio de recepción móvil y handheld gracias al time interleave y a la recepción parcial que se genera en su proceso de transmisión.

- Robustez contra la interferencia

La adopción de corrección de errores con interleave plural permite tener un sistema efectivo contra el ruido de impulso (ruido urbano).

En la figura 1.10 se observa de manera simplificada, la estructura del sistema de transmisión.

[9]

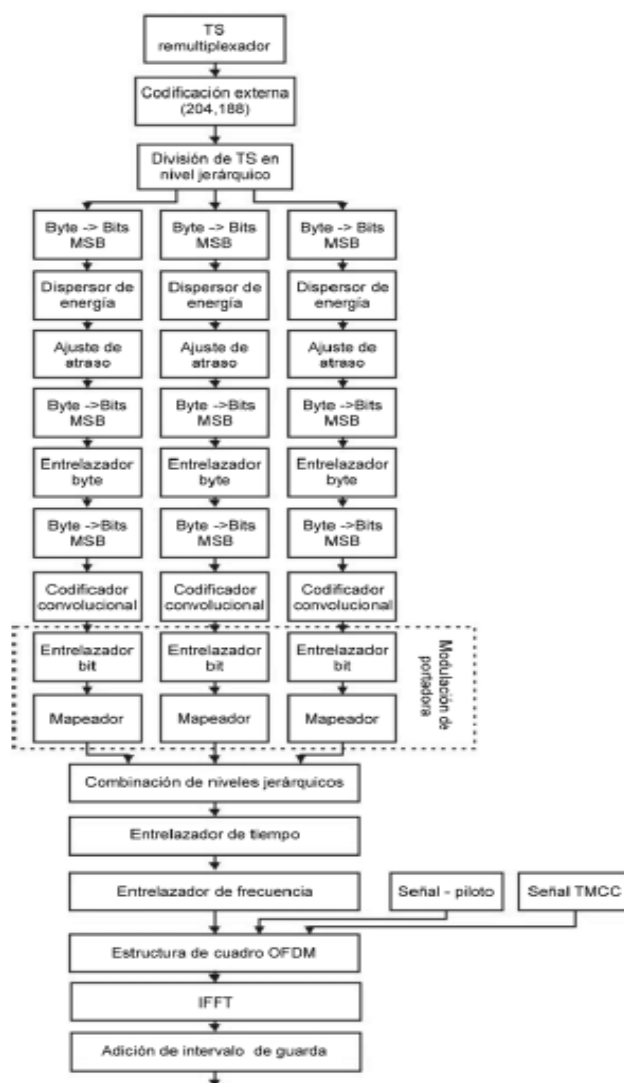


Figura 1.10 Diagrama de bloques del sistema de transmisión del estándar ISDB-Tb

1.2.4.1 Remultiplexación del Transport Stream

El re-multiplexor es la primera etapa del sistema de transmisión, este es el responsable de recibir el flujo de datos y decidir correctamente en cuál de las tres capas jerárquicas debe ser transmitido un paquete. El flujo de entrada puede ser un TS MPEG-2 (flujo formado por paquetes de 188 bytes) o por un BTS (flujo formado por paquetes de 204 bytes).



Figura 1.11 Transport Stream re multiplexado (modo 1, intervalo de guarda 1/8)

Una remultiplexación del TS debe obligatoriamente ser formada por cuadros múltiples como unidades elementales, cada cual consistiendo en un número n de paquetes TS. El número de paquetes TS usados para distintos modos de transmisión y distintas razones de intervalo de guarda debe estar de acuerdo con la Tabla 1.2

Tabla 1.2 Configuración de la multiplexación del cuadro multiplex

Modo	Numero de paquetes TS transmitidos dentro de un cuadro multiplex			
	Tasa intervalo de guarda 1/4	Tasa intervalo de guarda 1/8	Tasa intervalo de guarda 1/16	Tasa intervalo de guarda 1/32
Modo 1	1280	1152	1088	1056
Modo 2	2560	2304	2176	2112
Modo 3	5120	4608	4352	4224

1.2.4.2 Codificación Externa

La codificación externa consiste en aplicar a cada paquete TS un código Reed Solomon (RS) abreviado. La codificación RS abreviada (204,188) es generada agregando 51 bytes 0x00h

en el comienzo de los datos del código RS (255, 239); al final esos 51 bytes se deben remover obligatoriamente.

El código RS abreviado (204, 188) permite corregir hasta 8 bytes erróneos ubicados aleatoriamente en un paquete de 204 bytes. La figura 1.12 muestra el paquete de datos TS protegido por codificación RS. El paquete de 204 bytes protegido con el código corrector de error es también conocido como BTS.

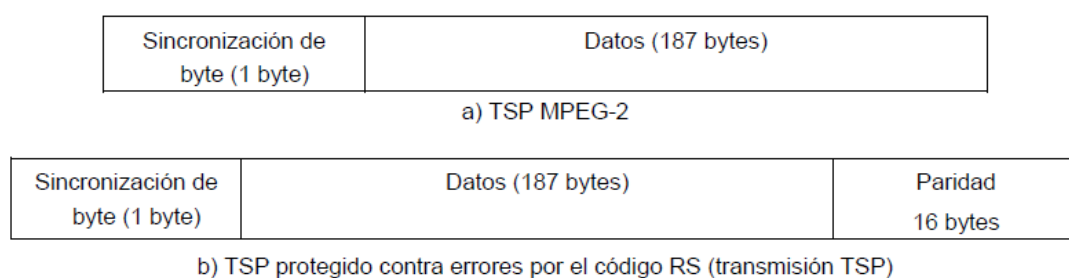


Figura 1.12 Paquete TS sin codificación RS(a) y con codificación RS(b)

1.2.4.3 División en Capas Jerárquicas

El divisor jerárquico divide al TS re multiplexado en porciones de 204 bytes (BTS) y asocia cada parte a la capa jerárquica específica. Al mismo tiempo, el divisor debe remover los paquetes nulos. La capa jerárquica a que pertenece la transmisión de paquetes TS es especificada por la información de la capa jerárquica basada en la organización. El número máximo de capas jerárquicas es tres.

1.2.4.4 Dispersión de Energía

La dispersión de energía se realiza para cada capa jerárquica. Se realiza mediante una operación OR-Exclusiva bit a bit entre el flujo de datos y una secuencia pseudo-aleatoria como se muestra en la figura 1.13. Esta operación garantiza una dispersión de energía del flujo de datos en el dominio de la frecuencia.

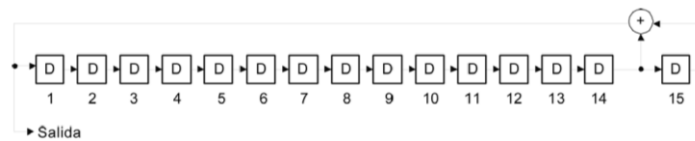


Figura 1.13 Generador de secuencias pseudo aleatorias

1.2.4.5 Ajuste de atraso

El ajuste de atraso, asociado al byte interleaving, tiene el objetivo de proveer tiempo de atraso igual para transmisión y recepción en todas las capas jerárquicas.

El montante de atraso de transmisión causado por el byte interleaving para una capa, puede diferir de otra, cuando se convierte en tiempo de atraso. Para compensar esta relativa diferencia en tiempo de atraso entre las capas jerárquicas, se realiza un ajuste para cada etapa, antes del byte interleaving, de acuerdo con la tasa de bit de transmisión.

1.2.4.6 Entrelazador de Byte

El entrelazador o byte interleaving que se usa en el sistema ISDB-Tb es de tipo convolucional y está compuesto por B ramas. Cada rama posee un determinado número de retrasos y el número de retrasos por rama sigue un orden determinado M. La primera rama tiene 0 retrasos, la segunda tiene M retrasos, la tercera rama tiene $2 \times M$ atrasos, y así sucesivamente, hasta llegar a la última rama que tiene $(B-1) \times M$ retrasos. El Entrelazador del estándar ISDB-Tb se muestra en la figura 1.14, y posee retrasos $M=17$ y ramas $B=12$.

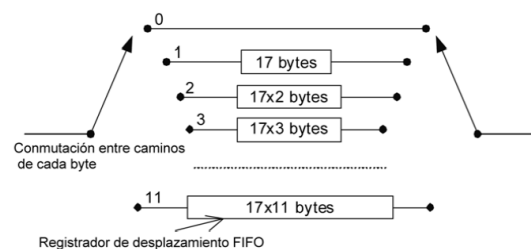


Figura 1.14 Circuito de byte interleaving

1.2.4.7 Codificación Interna

Se usa un código convolucional con punzado basado en un código convolucional madre de profundidad $k = 7$ y tasa de codificación $1/2$ como se muestra en la figura 1.15.

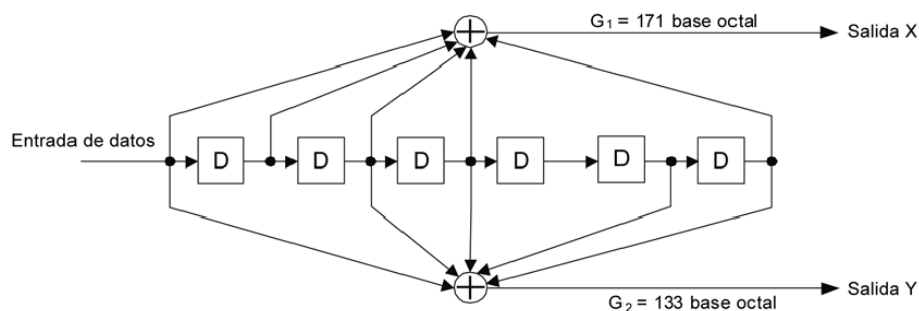


Figura 1.15 Circuito de codificación de código convolucional

Es posible configurar una tasa de codificación con diferentes valores, aumentando la tasa de código y disminuyendo el número de bits enviados reduciendo la protección contra errores. La tasa de codificación seleccionable del código interno debe estar de acuerdo a la tabla 1.3.

Tabla 1.3 Tasa del código interno y secuencia de la señal de transmisión

Tasa de código γ_c	Patrón de Punteo	Secuencia de Transmisión
1/2	X: 1 Y: 1	X1 Y1
2/3	X: 1 0 Y: 1 1	X1 Y1 Y2
3/4	X: 1 0 1 Y: 1 1 0	X1 Y1 Y2 X3
5/6	X: 1 0 1 0 1 Y: 1 1 0 1 0	X1 Y1 Y2 X3 Y4 X5
7/8	X: 1 0 0 0 1 0 1 Y: 1 1 1 1 0 1 0	X1 Y1 Y2 Y3 Y4 X5 Y6 X7

1.2.4.8 Entrelazador de bit

Los bits se agrupan al entrelazador dependiendo el mapeamiento seleccionado. Cada uno de estos entrelazadores tienen una estructura que se puede ver en la Figura 1.16, el bit más significativo es encaminado a la rama sin atraso. [2]

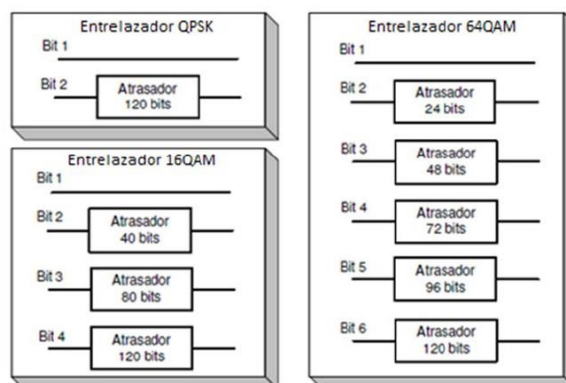


Figura 1.16 Estructura de los entrelazadores de bits

1.2.4.9 Modulación de la portadora

La modulación permite convertir el flujo de datos en ondas electromagnéticas para la transmisión de la señal. El mapeamiento se realiza dependiendo de la modulación escogida. El sistema ISDB-Tb permite 4 tipos de modulaciones y estas se escogen dependiendo del servicio que se vaya a ofrecer:

- QPSK (Quadrature Phase Shift Keying). La secuencia del flujo de datos procedente del codificador interno se divide en canales paralelos (I y Q). Permite transmitir dos bits por símbolo. SU diagrama de constelación se observa en la figura 1.17

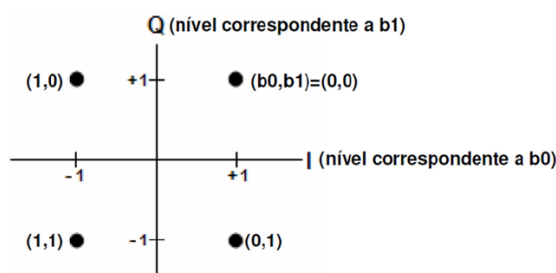


Figura 1.17 Constelación QPSK

- DQPSK (Differential Quadrature Phase Shift Keying). Permite transmitir la información en la diferencia de fase entre el símbolo actual y el símbolo anterior. A partir de información de 2 bits permite tener 8 posibles símbolos.

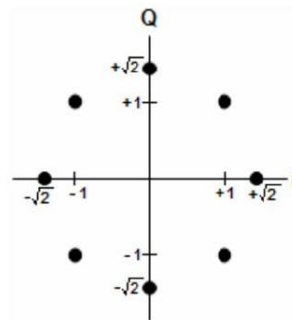


Figura 1.18 Constelación DQPSK

- 16QAM (16 Quadrature Amplitude Modulation) Formado por 16 símbolos, utiliza fase y amplitud de la portadora para transmitir 4 bits por símbolo. Su diagrama de constelación se muestra en la Figura 1.19.

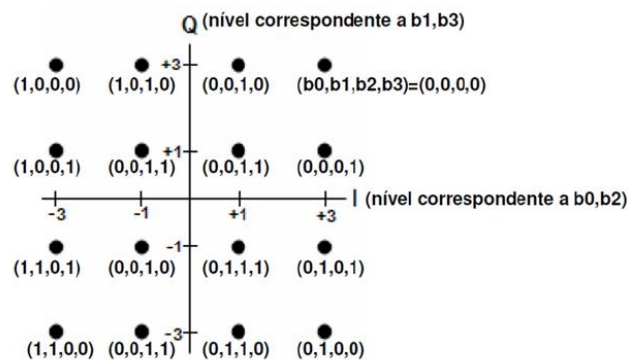


Figura 1.19 Constelación 16QAM

- 64QAM (64 Quadrature Amplitude Modulation). Guarda información tanto en fase como en amplitud de su portadora, para transmitir 6 bits por símbolo. Su diagrama de constelación se muestra en la Figura 1.20.

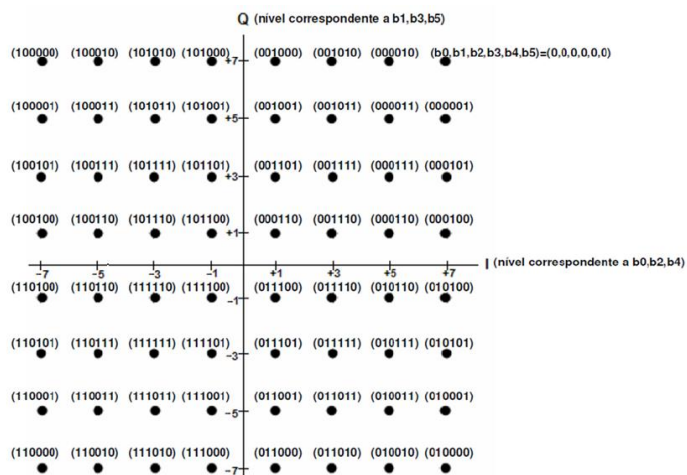


Figura 1.20 Constelación 64QAM

1.2.4.10 Combinación de capas jerárquicas

Las señales de las diferentes capas jerárquicas, una vez que han sido sometidas a la codificación interna y modulación de portadora son combinadas en insertadas en un segmento de datos y sometidas a la conversión de velocidad.

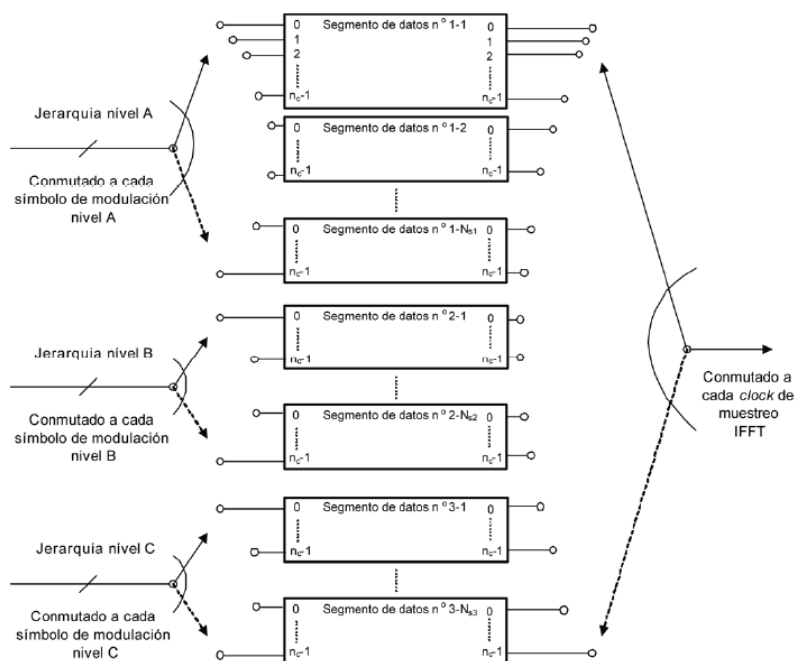


Figura 1.21 Combinador capas jerárquicas

Los datos son organizados para formar los segmentos, los símbolos complejos resultantes del proceso de mapeamiento de las diferentes capas, son combinados para entregar los datos en el orden necesario y generar los símbolos OFDM. [2]

1.2.4.11 Entrelazador de tiempo

Una vez que las capas jerárquicas se han combinado, se entrelazan en el tiempo en unidades de símbolos de modulación para cada uno de los ejes I y Q.

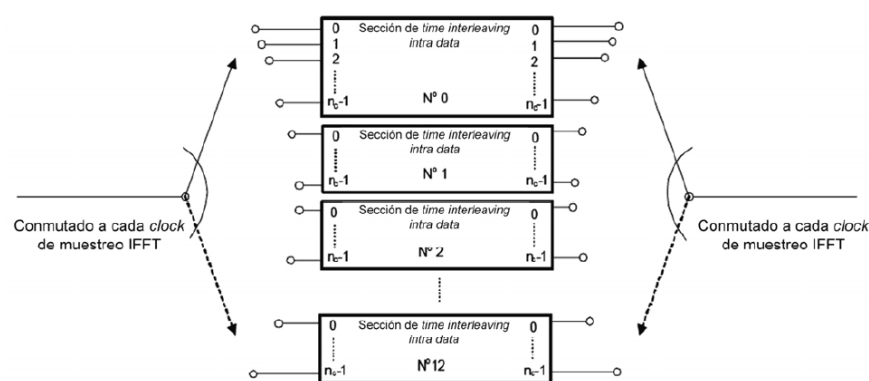


Figura 1.22 Configuración de la sección de entrelazado de tiempo

El entrelazador de tiempo combina los símbolos mapeados a través de la inserción de atrasos en cada símbolo, de forma que una secuencia de datos no se transmita al mismo tiempo, aunque estén en portadoras diferentes. Esta acción aumenta la robustez contra el desvanecimiento (fading). La Figura 1.23 muestra un ejemplo del efecto de utilizar entrelazador de tiempo.



Figura 1.23 Ejemplo entrelazamiento de tiempo sobre un ruido impulsivo

1.2.4.12 Entrelazador de frecuencia

El entrelazado de frecuencia combina las portadoras de los distintos segmentos con el fin de aleatorizar el espectro de las frecuencias y reducir los efectos destructivos del canal en la señal generada. Cuando ocurre un desvanecimiento de multi-trayecto este alcanza a portadoras de diferentes segmentos, el entrelazado de frecuencia dispersa este error generado.

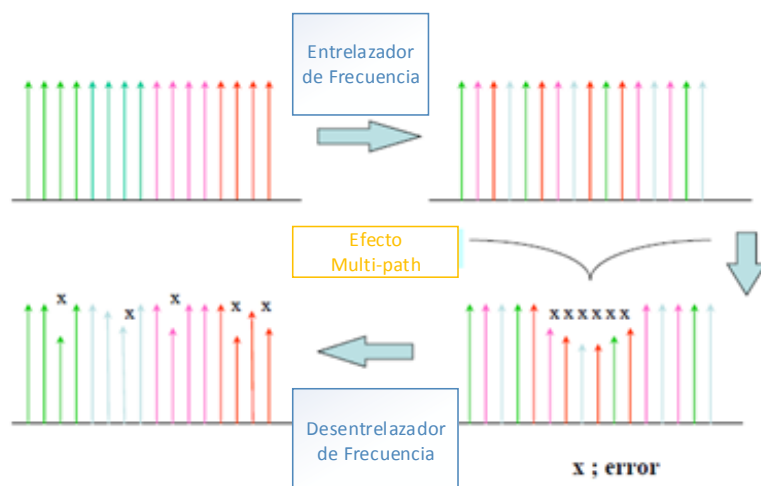


Figura 1.24 Efecto del entrelazamiento de frecuencia

1.2.4.13 Señal OFDM

El sistema de transmisión del estándar ISDB-Tb utiliza la segmentación de banda con técnicas OFDM, las cuales envían múltiples portadoras ortogonales en el mismo canal. Es decir al usar la Transformada Rápida de Fourier Inversa (IFFT) para generar los símbolos, en la posición adyacente de la portadora las energías de las otras portadoras son cero, esto lo demuestra la figura 1.25

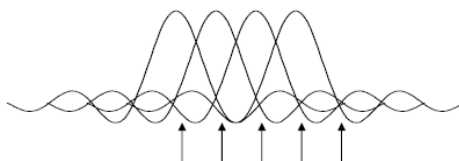


Figura 1.25 División ortogonal multiplexada

Un ejemplo de la forma de onda de la señal OFDM transmitida se observa en la figura 1.26

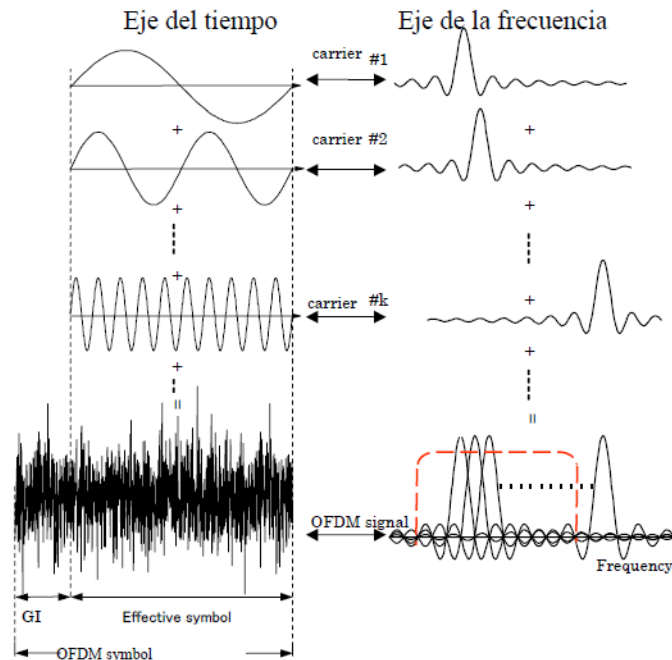


Figura 1.26 Señal OFDM

El símbolo OFDM es formado por portadoras que transportan los datos de la señal de televisión, más portadoras especiales que contienen información adicional del sistema. Las portadoras especiales son: SP (Scattered Pilot), CP (Continual Pilot), AC's (Auxiliary Channel) y TMCC (Transmission and Multiplexing Configuration Control).

El tiempo de guarda, el cual consiste en hacer una copia de la parte final del símbolo OFDM al inicio del mismo símbolo, permite tener mayor robustez frente a efectos de multi-trayectos e interferencia intersimbólica. El sistema ISDB-Tb permite configurar la duración del intervalo de guarda seleccionando una parte de tiempo del símbolo OFDM efectivo. Las opciones disponibles son 1/4, 1/8, 1/16 y 1/32. La tabla 1.4 muestra la duración de símbolo y cuadro OFDM. Una señal para ser transmitida debe ser organizada en cuadros, cada cuadro consisten 204 símbolos OFDM.

Tabla 1.4 Duración de símbolo y cuadro OFDM

MODO	1			
Tiempo efectivo de símbolo (us)	252			
Intervalo de guarda	1/4	1/8	1/16	1/32
Duración tiempo de Guarda (us)	63	31.5	15.76	7.32
Duración de símbolo OFDM (us)	315	283.5	267.75	259.875
Duración de cuadro OFDM (ms)	64.26	57.83	54.621	53.0145
MODO	2			
Tiempo efectivo de símbolo (us)	504			
Intervalo de guarda	1/4	1/8	1/16	1/32
Duración tiempo de Guarda (us)	126	63	31.5	15.76
Duración de símbolo OFDM (us)	630	567	533.5	519.75
Duración de cuadro OFDM (ms)	128.52	115.668	108.242	106.029
MODO	3			
Tiempo efectivo de símbolo (us)	1008			
Intervalo de guarda	1/4	1/8	1/16	1/32
Duración tiempo de Guarda (us)	252	126	63	31.5
Duración de símbolo OFDM (ms)	1.26	1.134	1.071	1.0395
Duración de cuadro OFDM (ms)	257.04	231.336	218.484	212.058

1.2.5 Capacidad de Transmisión del Sistema

La capacidad de tasa útil de transmisión varía de acuerdo a las configuraciones seleccionadas en cada capa jerárquica. El sistema ISDB-Tb está compuesto por 13 segmentos, por lo que tenemos que calcular la velocidad de transferencia en un segmento y después multiplicarlo por los segmentos de cada grupo. Así, se obtiene la velocidad de transferencia de la capa jerárquica. Si se configuraron las 3 capas A, B y C se realiza el mismo procedimiento y se suma el total.

La siguiente ecuación nos permite calcular la capacidad máxima en la tasa de transmisión en 1 segmento de acuerdo a los parámetros configurados:

$$R_i = \frac{8 m r_{RS} r_C}{21 (1+\alpha)} \quad (1.1)$$

Dónde:

- m = hace referencia a la modulación de la capa la cual puede tomar los valores 2 (QPSK y DQPSK) 4 (16QAM) o 6 (64QAM)
- r_{RS} = valor constante de 188/204. Este valor tiene relación con los 16 últimos bytes que sea integran con codificación Reed Solomon
- r_C = Hace referencia a la codificación interna aplicada a la capa jerarquica la cual puede tomar los valores de 1/2, 2/3, 3/4, 5/6, 7/8.
- α = al tiempo de guarda aplicado a la transmisión. Puede tomar los valores de 1/4, 1/8, 1/16, 1/32.

1.2.6 Tarjetas Moduladoras ISDB-Tb

En el mercado internacional se puede encontrar varias marcas que ofrecen moduladoras para el sistema ISDB-Tb. Existen moduladoras que son multiestándar, algunas tienen los bloques de codificación, multiplexación y transmisión embebidas, otras solo contemplan el bloque de transmisión. A continuación se presenta algunas moduladoras que se ofertan en el mercado internacional. La tarjeta DekTec DTU-215, tarjeta moduladora con la que se desarrolla el presente proyecto, se especifica en el capítulo 2.

1.2.6.1 Kathrein NDS-3542 Encoder Modulator ISDB-T

NDS-3542 es un encoder y modulador que integra codificación de video MPEG2 HD/MPEG4 HD y modulación que soporta el estándar ISDB -T (DVB-C/-T/ATSC Opcional) permitiendo convertir señales HDMI en señales de Radio Frecuencia (RF) ISDBT-T. NDS-3542 está equipado con los siguientes puertos:

- 2 canales de entrada HDMI.
- 1 puerto de entrada ASI, 2 puertos de salida ASI.
- 1 puerto de salida UDP IP, 1 puerto de salida RF.



Figura 1.27 Kathrein NDS-3542

La señal fuente puede ser de un receptor de satélite, reproductor Blue-ray, cámaras de circuito cerrado de televisión, antena, etc. [10]

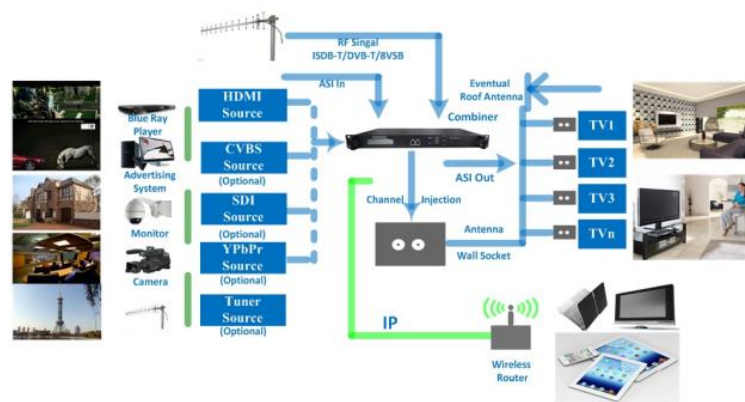


Figura 1.28 Aplicaciones típicas Kathrein NDS-3542

1.2.6.2 PROMAX MO-370 LE

El MO-370 LE es un modulador ISDB-Tb de propósito general construido sobre un chasis 1U de 19". Dispone de dos entradas serie MPEG TS-ASI las cuales se pueden utilizar para modular señal COFDM. Una señal adicional test TS puede ser generada internamente. Además cuenta con un canal de salida RF que trabaja en el rango de frecuencias 45~875MHz.

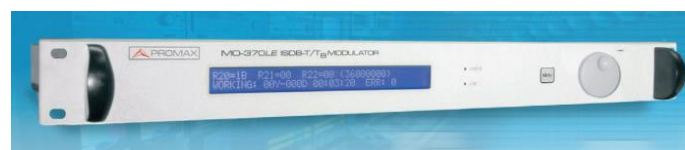


Figura 1.29 PROMAX MO-370 LE

El modulador puede ser configurado para generar cualquiera de los modos de transmisión descritos por el estándar ISDB-T/TB. [11]

1.2.6.3 TECSYS TS 9500-ISDB-T

Indicado para la transmisión de televisión digital terrestre ISDB-T. Posee dos entradas ASI que puede recibir TS o BTS. Tiene un Oscilador de alta precisión de 10 MHz integrado con opción de entrada de 100 MHz externo. Tiene una salida IF de 44 MHz con pre-corrector digital. Totalmente desarrollado en Brasil, es montado en un chasis de aluminio de 19'' con unidad de ventilación activa y una interfaz de usuario amigable con panel frontal y display LCD. [12]



Figura 1.30 TECSYS TS 9500-ISDB-T

1.2.6.4 DekTec DTA-115

La tarjeta DTA-115 es un modulador multi estándar con VHF/UHF upconverter. Soporta las modulaciones QAM, OFDM y VSB que mantienen los estándares DVB, ATSC, ADTB e ISDB-T. Sus canales trabaja en el rango de frecuencias 47~862MHz. En el año 2011, la ESPE implemento un transmisor de pruebas de TDT con el estándar ISDB-Tb, la etapa de modulación se realizó con esta tarjeta.



Figura 1.31 DekTec DTA-115

CAPÍTULO II

2. PLATAFORMA DE TRANSMISIÓN DEKTEC

DekTec Digital Video es una empresa holandesa que diseña y manufactura tarjetas de extensión para PC, dispositivos USB/IP y software para el mercado profesional de televisión digital. Sus productos pueden ser usados en ambientes de laboratorio para pruebas y mediciones, profesionalmente para crear una infraestructura broadcast o como fabricante del equipamiento original de un componente en una solución. [13]

La plataforma de transmisión DekTec para el sistema ISDB-Tb, cuenta con su software propietario StreamXpress que permite reproducir TS o BTS, y una tarjeta moduladora, las cuales pueden ser DTA-2111, DTA-2115, DTA-111, DTA-115 o DTU-215 con sus distintas prestaciones. Actualmente, la Universidad de las Fuerzas Armadas – ESPE cuenta con distintos productos de la marca DekTec, los cuales son usados en sus proyectos de investigación y vinculación con la sociedad.

2.1 TARJETA MODULADORA DEKTEC DTU-215

En el presente proyecto se trabaja con la tarjeta moduladora DTU-215, la cual fue puesta en el mercado en el año 2010. La tarjeta DTU-215 es considerada la cumbre en su campo dado que es una moduladora VHF/UHF integrada en un chasis pequeño y versátil, es alimentado por el puerto USB y soporta el amplio rango de modulaciones de los distintos estándares de Televisión Digital.

2.1.1 Características Generales

La moduladora presenta las siguientes características:

- Soporta todas las constelaciones y modos de modulación para los estándares ISDB-T, DVB-T/T2/C/C2, ATSC, DTMB.
- Permite la reproducción de archivos de muestreo I/Q.
- Atenuador programable.
- Es compatible con el Software Development Kit (SDK) de Windows y Linux.
- Es alimentado por el puerto USB-2, por lo que no se necesita una fuente de energía externa.

Tabla 2.1 Atributos Técnicos DTU-215

Parámetros		Valores
Conector RF		75-Ω Hembra
Rango de Frecuencias		36 .. 1002MHz ±3ppm
Ancho de Banda (máx.)		8.0MHz
Tasa de muestreo I/Q		4.7 .. 9.375MHz
Nivel	Rango	-46 .. -15dBm (QAM) -49 .. -18dBm (OFDM)
	Tamaño de escalón	0.5dB
	Exactitud	±2dB
Canal Adyacente		-54dB (QAM) -52dB (OFDM)
Ruido de Fase		<-95dBc @ 10kHz
Pureza Espectral		>50dB (47 .. 1000MHz)
Puerto USB		USB-2
Energía (desde USB-2)		5V, 500mA
Dimensiones (LxWXH)		123 x 62 x 22mm

La tarjeta DTU-215 es utilizada para desarrollo e investigación, demostraciones y como generador de pruebas de los distintos estándares.



Figura 2.1 DekTec DTU-215

2.2 SOFTWARE DEKTEC STREAMXPRESS

DTC-300-SP StreamXpress es un paquete de software en Windows diseñado para realizar la reproducción a tiempo real de TS compatibles y archivos Standard Definition – Serial Digital Interface (SD-SDI). StreamXpress fue diseñado para ser ejecutado en una PC o Laptop y trabajar en conjunto con un dispositivo DekTec.

El presente software permite leer un TS o archivo SD-SDI del disco duro local y conectarlo a la salida de un dispositivo DekTec permitiendo su reproducción a tiempo real sobre una interface Asynchronous Serial Interface (ASI), RF o SD-SDI. Igualmente, StreamXpress provee de información del TS como contenido, identificación de paquetes e información del archivo. [14]. El proyecto actual abarca el uso de un modulador que permite trabajar con distintos estándares; sin embargo, únicamente va a ser usado en el sistema ISDB-Tb. De igual forma, a pesar que StreamXpress soporta varios estándares de Televisión Digital, en esta sección se presenta características enfocadas en el sistema ISDB-Tb.

2.2.1 Características Generales

Dependiendo el tipo de hardware que se conecta a StreamXpress, el programa mostrará mayor o menor número de prestaciones. A continuación se detalla la interfaz de usuario al conectar la tarjeta DTU-215.

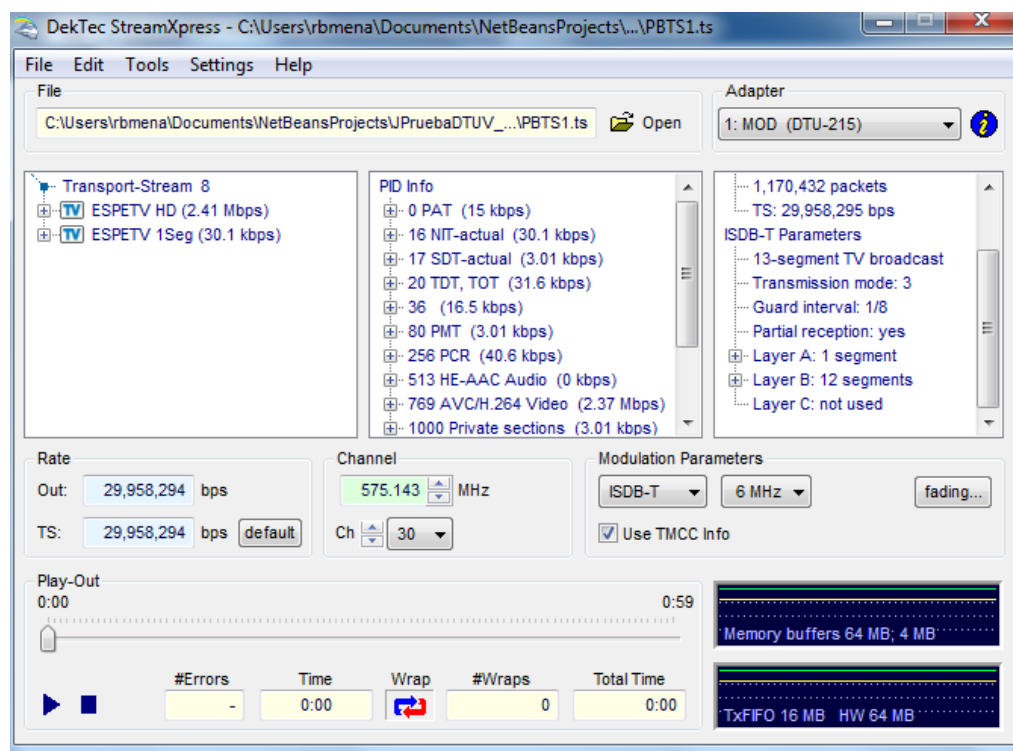


Figura 2.2 GUI StreamXpress

Tabla 2.2 Detalle componentes generales de la interfaz gráfica de StreamXpress

#	Variable	Descripción	Tipo
1	Adaptador	Selecciona el adaptador y puerto de reproducción	Desplegar
2	Estado Reproducción	Indica la longitud del TS basado en su Program Clock Reference (PCR) y tasa de bits. Además con el cursor indica el momento del proceso en la reproducción	Display
3	Archivo	Indica el nombre y ubicación del TS	Display
4	Reproducir/Pausar/Parar	Controla el botón de Reproducir, Pausar o Parar la transmisión del TS. Al presionar botón Parar se transmite paquetes nulos	Botón
5	Error	Contador interno que se incrementa cuando se pierden paquetes debido al desbordamiento de memoria causado por TS mal configurados o limitación de recursos en la PC	Display
6	Tiempo	Representa tiempo de reproducción desde el inicio de TS	Display
7	Repetir	Cuando es presionado, el TS se reproduce infinitamente.	Botón
8	Número de Reproducciones	Contador que muestra el número de veces que se ha transmitido el TS	Display
9	Tiempo Total	Representa tiempo de reproducción total desde la última vez que se presionó el botón Parar	Display

10	Buffer Memoria	Indica el tamaño y variación en el tiempo de la memoria buffer del software. Si cae a cero, el buffer está vacío y la reproducción parará	Display
11	Buffer Hardware	Indica el tamaño y variación en el tiempo de la memoria buffer del dispositivo DekTec usado. Si cae a cero, la reproducción parará	Display
12	Canal	Permite seleccionar el canal RF reasignado a una frecuencia de acuerdo al estándar.	Desplegar
13	Ancho de Banda	Establece el ancho de banda del canal RF	Desplegar
14	Usar información TMCC	Aparece si un archivo BTS con información TMCC es abierto. Al seleccionar se transmite el archivo con los datos ya configurados	Check
15	Parámetros	Abre nueva ventana donde se configura los parámetros de las capas y modulación.	Botón

De igual importancia que las características ya expuestas, al abrir un archivo TS, StreamXpress presenta al usuario información importante del contenido del mismo. Existen tres áreas que indican los contenidos.

✓ Área de información del TS

Indica el identificador del TS. Además de identificar cada programa dentro del stream, muestra el nombre del programa y su tasa de transmisión. Al desplegar cada programa, indica el número de PID, tipo de stream, identificador de stream y bitrate de cada componente.

✓ Área de información PID

Indica cada PID y su asociación con un programa, el bitrate del PID, si existe presencia del PCR, Identificador del stream

✓ Área de información del archivo

Indica el nombre del archivo, su tamaño en bytes, numero de paquetes en el archivo, el tamaño de cada paquete. Si un archivo BTS es abierto, se muestra la información de las capas y todos los parámetros de modulación.

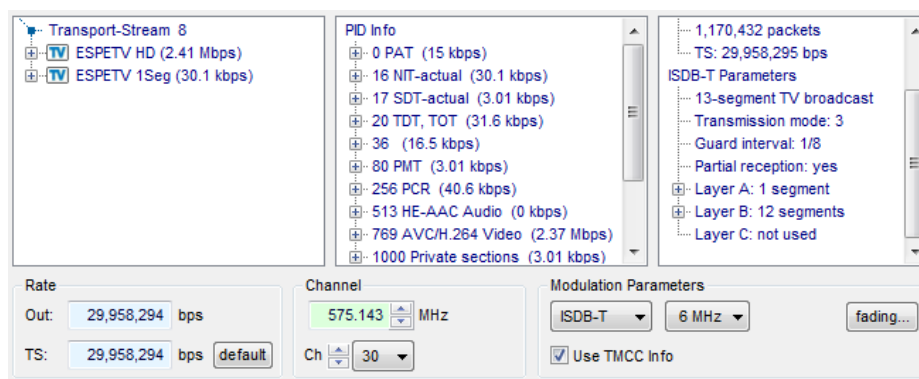


Figura 2.3 Áreas de información del TS

2.2.2 Transmisión de TS con el estándar ISDB-Tb

Los parámetros de las capas jerárquicas y modulación se configuran en una ventana que aparece al presionar el Botón de Parámetros. A continuación se detalla los componentes de la ventana de parámetros ISDB-Tb.

2.2.2.1 Mapeamiento del TS en las capas jerárquicas

Tabla 2.3 Detalle de la sección de mapeamiento de capas jerárquicas

#	Variable	Descripción	Tipo
1	PID/SVC/Bitrate	Muestra información del TS como el PID, service component type y tasa de transmisión	Display
2	Capas Activas	Permite al usuario seleccionar en que capa jerárquica (A, B o C) asignar un programa o servicio	Selección
3	Mapear PID a capa	Si un programa o servicio no ha sido asignado, el usuario escoge una capa jerárquica como asignación por default	Desplegable
4	Permitir PIDS en múltiples capas	Si está activo, permite que un servicio o programa sea usado en múltiples capas	Check

PID	Svc/Comp	Rate	A	B	C
0x0000	PAT	15,040	-	B	-
0x0010	NIT-actual	30,080	A	-	-
	ESPETV HD	2,413,920			
0x0301	AVC/H.264 Video	2,367,296	-	B	-
0x03e8	Private Sections	3,008	-	B	-
	ESPETV 1Seg	30,080			
0x0011	SDT-actual	3,008	A	-	-
0x0014	TDT, TOT	31,584	A	-	-
0x0024		16,544	A	-	-
0x0050	PMT	3,008	-	B	-
0x0100	PCR	40,608	-	B	-

Map other PIDs to layer: -- ▾ Allow PID in multiple layers

Figura 2.4 Sección de mapeamiento de capas jerárquicas

2.2.2.2 Parámetros ISDB-Tb

Esta sección permite la selección de los parámetros del estándar ISDB-Tb. En la figura 2.5 y tabla 2.4 se detalla la presente sección.

Tabla 2.4 Detalle de la sección parámetros ISDB-Tb

#	Variable	Descripción	Tipo
1	Tipo de transmisión	Permite seleccionar el tipo de transmisión. Estas pueden ser Televisión (13 segmentos), Radio 1-Segmento o Radio 3-Segmentos	Desplegable
2	Modo	Permite seleccionar el modo de transmisión, los cuales definen el número de portadoras por segmento. Estos pueden ser modo 1 (108), modo 2 (216) o modo3 (432).	Desplegable
3	Intervalo de guarda	Permite seleccionar el intervalo de guarda de la transmisión OFDM. Estos pueden ser 1/4, 1/8, 1/16, 1/32.	Desplegable
4	IIP PID	Establece el PID usado para transferir la ISDB-Tb Information Packet (IIP) al modulador. IIP contiene la información TMCC	Campo
5	Recepción Parcial	Si está activo, permite el uso de transmisión one-seg y la capa A debe tener un segmento dedicado para esto	Check
6	Transmisión de Emergencia	Si está activo, se habilita la bandera de alarma de emergencia (1 bit) en la información TMCC	Check

The image shows a software interface titled "ISDB-T Parameters". It contains several configuration options:

- Broadcast Type:** A dropdown menu set to "Television".
- Mode:** A dropdown menu set to "3".
- Guard:** A dropdown menu set to "1/8".
- IIP PID:** A text input field containing "0x0000".
- Partial Reception:** A checked checkbox.
- Emergency Broadcasting:** A checked checkbox.

Figura 2.5 Sección de parámetros ISDB-Tb

2.2.2.3 Parámetros de la Capa Jerárquica

Esta sección permite la selección de los parámetros del estándar de las capas jerárquicas. En la figura 2.6 y tabla 2.5 se detalla la presente sección.

Tabla 2.5 Detalle de la sección parámetros de capa jerárquica

#	Variable	Descripción	Tipo
1	Numero de Segmentos	Permite seleccionar el número de segmentos asignado a las capas jerárquicas	Desplegable
2	Modulación	Permite seleccionar el tipo de modulación para cada capa jerárquica. Las modulaciones pueden ser DQPSK, QPSK, 16QAM o 64QAM	Desplegable
3	Tasa de código interno	Permite seleccionar la tasa de código interno para cada capa jerárquica. Las tasas pueden ser 1/2, 2/3, 3/4, 5/6, 7/8.	Desplegable
4	Tiempo de entrelazado	Permite seleccionar el tiempo de entrelazado. Este permite tener una señal más robusta y depende del modo de transmisión escogido. El tiempo puede ser 0, 4, 8, 16 (modo 1); 0, 2, 4, 8 (modo2); 0, 1, 2, 4 (modo 3).	Desplegable
5	Tasa de transmisión (bps)	Este campo muestra la tasa de transmisión permitida en cada capa calculado en base a las configuraciones previas	Display
6	Tasa de Transmisión Seleccionada (bps)	Este campo muestra la tasa de transmisión seleccionada de los distintos programas o servicios que han sido asignados a la capa	Display
7	Total	Indica el total de segmentos asignados a las capas jerárquicas. El valor debe ser 13.	Display

Layer Parameters						
	#Segments	Modulation	Code Rate	Time Intlv	Rate (bps)	Selected (bps)
A	1	QPSK	5/6	I = 2	520,109	111,296
B	12	16QAM	3/4	I = 4	11,234,360	2,428,960
C	0	64QAM	3/4	I = 2	0	0
Total	13				11,754,469	2,540,256

Figura 2.6 Sección de parámetros de capa jerárquica

2.2.2.4 Centro de Mensajes

Esta sección muestra cualquier error generado al configurar los distintos parámetros como muestra la figura 2.7.

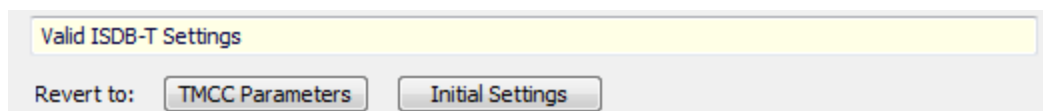


Figura 2.7 Sección Centro de mensajes

Tabla 2.6 Detalle de la sección Centro de mensajes

#	Variable	Descripción	Tipo
1	Centro de mensaje	Indica si la configuración es válida y muestra cualquier error	Display
2	Parámetros TMCC	Si es posible, carga los parámetros TMCC presentes en un BTS	Botón
3	Configuración Inicial	Carga la configuración inicial de la modulación	Botón

2.2.3 Requerimientos de Hardware

Dependiendo de la aplicación y el dispositivo DekTec utilizado, los requerimientos pueden variar.

Sistema Operativo:

Microsoft Windows XP/2K3/Vista/2K8/7

Procesador y memoria RAM:

III* 300 MHz y 256MB para reproducción ASI

Core 2* with 2GB of RAM para reproducción DVB-T2

*O procesador AMD equivalente

Unidad de Disco Duro

Tasas combinadas > 50Mb/s usa disco separado del sistema operativo

Tasas combinadas \leq 100Mb/s usa disco SATA 7200 rpm

Tasas combinadas > 100Mb/s usa disco SATA 10000 rpm

2.3 INTERFAZ DE PROGRAMACIÓN DE APLICACIONES DEKTEC

La Interfaz de Programación de Aplicaciones DekTec o DTAPI (por sus siglas en inglés) es el conjunto de métodos que permite a aplicaciones de software acceder a las funciones de los dispositivos DekTec; en otras palabras, DTAPI permite manejar un dispositivo DekTec en un nivel más alto de abstracción, de lo que sería posible usando llamadas directas de los controladores del dispositivo. [15]

Esta API desarrollada por DekTec permite desarrollar aplicaciones a necesidad del usuario, accediendo a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación.

Es importante mencionar que DTAPI permite acceder a toda la gama de productos DekTec y sus diferentes funciones. En el caso de la tarjeta moduladora DTU-215, que tiene funciones limitadas en comparación de otros productos, solo se utilizará ciertos métodos definidos en la DTAPI. Por lo tanto la descripción a continuación representa únicamente los utilizados en el presente proyecto.

2.3.1 Descripción General

DTAPI es la colección de clases C++ donde cada una representa una función genérica de un dispositivo DekTec. Las instancias de una clase DTAPI refiere a una función específica de un dispositivo en particular, en este caso la tarjeta moduladora DTU-215 los cuales al invocarlos controlan indirectamente al dispositivo.

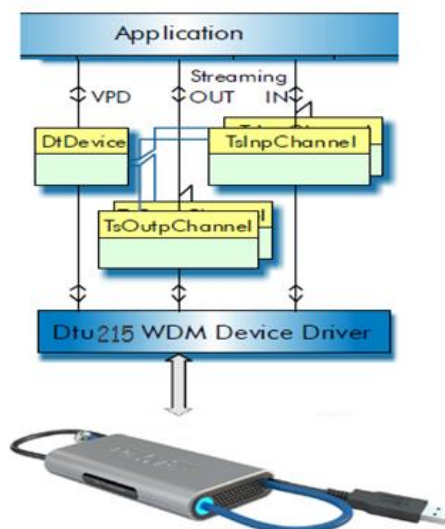


Figura 2.8 Ejemplo de Objetos DTAPI sobre moduladora DTU-215

La figura 2.8 ejemplifica un número de objetos DTAPI inicializados en una aplicación. La aplicación interactúa con los objetos DTAPI. Los métodos de la librería que implementan las clases DTAPI se comunican a la tarjeta moduladora a través de los controladores del dispositivo.

Cada dispositivo DekTec es representado por un objeto DtDevice el cual es el punto de inicio para interactuar con el mismo. Funciones como inicialización de canales para la transmisión, obtener información del controlador y lectura/escritura de su Vital Product Data (VPD) soporta este objeto. Funciones de transmisión de MPEG-2 TS son representados por objetos de canal DtOutputChannel. En la siguiente sección se detalla características, funcionamiento y sintaxis de las clases y estructuras utilizados en el la plataforma de transmisión.

2.3.2 Contenido de Clases ISDB-Tb

En informática, una clase es una plantilla que permite la creación de objetos de datos que tiene características y comportamientos comunes. Cada clase define un conjunto de variables y métodos que operan similarmente. En la presente sección se estudia las clases DTAPI usadas en el presente proyecto.

2.3.2.1 Clase DtDevice

Define elementos del dispositivo, permite interactuar con él y es la puerta de entrada hacia la manipulación de las funcionalidades del mismo.

✓ DtDevice::AttachToType

Permite vincular el dispositivo a la aplicación desarrollada.

Tabla 2.7 DtDevice::AttachToType

DTAPI_RESULT DtDevice::AttachToType (int, int);	
[in] int TypeNumber,	Valor entero que representa el número del dispositivo que se va a vincular. En este caso es 215.
[in] int DeviceNo=0	Si el sistema contiene distintos dispositivos conectados simultáneamente. Comienza en 0.

✓ DtDevice::Detach

Desvincula el dispositivo de la aplicación desarrollada

Tabla 2.8 DtDevice::VpdRead

DTAPI_RESULT DtDevice::VpdRead (const char*, char);	
[in] const char* <i>pKeyword</i> ,	Cadena de caracteres que identifica el atributo VPD a ser leído. Puede leerse el numero serial, numero de parte, identificador de cliente, etc.
[out] char* <i>pVpdItem</i>	Cadena de caracteres recibido de la memoria EEPROM

Tabla 2.9 DtDevice::Detach

DTAPI_RESULT DtDevice::Detach();	
void	.

✓ DtDevice::VpdRead

Lee el Vital Product Data de la memoria EEPROM del dispositivo

2.3.2.2 Clase DtIsdbtPars

Esta clase describe los parámetros para la modulación ISDB-Tb

Tabla 2.10 class DtIsdbtPars

class DtIsdbtPars { ... }	
bool <i>m_DoMux</i> ;	Si es verdadero permite multiplexación jerárquica. Caso contrario los parámetros de modulación se especifican en la información TMCC de los 16 bytes extras en los paquetes de 204.
cbool <i>m_FilledOut</i> ;	Si la multiplexación jerárquica está activa, indica que todos los parámetros como intervalo de guarda, modo, modulación, etc. han sido configurados.
int <i>m_ParXtra0</i> ;	Este parámetro es codificado como ParXtra0 en el método SetModControl con <i>ModType</i> DTAPI_MOD_ISDBT
int <i>m_BType</i> ;	Se debe especificar el valor de DTAPI_ISDBT_BTTYPE_TV para transmisión de televisión con cualquier número de segmentos
int <i>m_Mode</i> ;	Se debe especificar el valor del modo de transmisión que se va a usar. Estos valores pueden ser 1, 2, 3
int <i>m_Guard</i> ;	Se debe especificar el valor del intervalo de guarda que se va a utilizar. Este valor está dado por DTAPI_ISDBT_GUARD_1_32 o todas sus variantes ya mencionadas previamente

<code>int m_PartialRx;</code>	Es una bandera que si está activa (1) indica que la capa jerárquica A es usada para recepción parcial.
<code>int m_Emergency;</code>	Es una bandera que si está activa (1) indica que el bit de control de transmisión de emergencia está activado
<code>int m_IipPid;</code>	Valor PID que es usado para la multiplexación de paquetes IIP.
<code>DtIsdbtLayerPars m_LayerPars[3];</code>	Vector donde se configura la modulación de las capas jerárquicas A (elemento 0), B (1) y C (2).
<code>std::map<int, int> m_Pid2Layer;</code>	Mapa que especifica a que capa jerárquica un stream es mapeado. Usa el PID para designar a que capa jerárquica en conjunto con el valor DTAPI_ISDBT_LAYER_A dependiendo la capa.
<code>int m_LayerOther;</code>	Captura todos los PIDs que no fueron mapeados y los envía a una capa jerárquica específica.
<code>bool m_Valid;</code>	Indica si los parámetros ISDB-T son válidos. Este es un parámetro derivado y es controlado por <code>DtIsdbtPars::CheckValidity</code> .
<code>int m_TotalBitrate;</code>	Bitrate en bps de todo el TS. El bitrate incluye los 16 bytes de extras de paridad.

✓ `DtIsdbtPars::ComputeRates`

Calcula la tasa de bits para cada capa jerárquica y almacena el resultado en el objeto que llama a esta función.

Tabla 2.11 `DtIsdbtPars::ComputeRates`

DTAPI_RESULT DtIsdbtPars::ComputeRates ();	
void	.

✓ DtIsdbtPars::CheckValidity

Valida si los parámetros ISDB-Tb fueron bien configurados. Su resultado es guardado en el booleano m_Valid.

Tabla 2.12 DtIsdbtPars::CheckValidity

DTAPI_RESULT DtIsdbtPars::CheckValidity (int&);	
[out] int& ResultCode	Indica el resultado de la validación. Si el valor es DTAPI_ISDBT_OK indica que los parámetros son válidos. Caso contrario indicara otros valores mostrando el error que se generó.

✓ DtIsdbtPars::RetrieveParsFromTs

Recupera los parámetros de modulación del BTS que contiene la información TMCC en los 16 bytes extras y almacena su resultado en el objeto DtIsdbtPars que llama a esta función.

Tabla 2.13 DtIsdbtPars::RetrieveParsFromTs

DTAPI_RESULT DtIsdbtPars::RetrieveParsFromTs (char* , int);	
[in] char* pBuffer,	Memoria buffer que contiene el BTS para poder recuperar los parámetros ISDB-Tb
[in] int NumBytes,	Numero de bytes de TS almacenado en el buffer

2.3.2.3 Clase DtOutputChannel

Es una clase que representa a un canal de salida para la transmisión de MPEG-2 Transport Stream.

✓ DtOutpChannel::AttachToPort

Vincula el objeto de canal de salida con un puerto físico específico.

Tabla 2.14 DtOutpChannel::AttachToPort

DTAPI_RESULT DtOutpChannel::AttachToPort (DtDevice*, int, bool);	
[in] DtDevice* pDtDvc,	Puntero hacia el objeto que representa el dispositivo DekTec. Este objeto debe estar vinculado al dispositivo.
[in] int Port,	Número del puerto físico. Por default el puerto es 1.
[in] bool ProbeOnly =false	Solo verifica si el canal ya está en uso.

✓ DtOutpChannel::ClearFifo

Limpia el contenido de la cola First In First Out (FIFO) y ubica el estado de control de transmisión en DTAPI_TXCTRL_IDLE.

Tabla 2.15 DtOutpChannel::ClearFifo

DTAPI_RESULT DtOutpChannel::ClearFifo ();	
void	

✓ DtOutpChannel::ClearFlags

Limpia las banderas que han sido enclavadas.

Tabla 2.16 DtOutpChannel::ClearFlags

DTAPI_RESULT DtOutpChannel::ClearFlags (int);	
[int] int Latched	Toma el valor de las banderas enclavadas. Las más importantes son DTAPI_TX_SYNC_ERR que muestra un error de sincronización en la transmisión y DTAPI_TX_FIFO_UFL que indica un error en desbordamiento de memoria.

✓ DtOutpChannel::Detach

Desvincula el objeto de canal de salida con un puerto físico específico.

Tabla 2.17 DtOutpChannel::Detach

DTAPI_RESULT DtOutpChannel::Detach (int);	
[in] int DetachMode	Toma un valor, que puede ser DTAPI_INSTANT_DETACH el cual desvincula inmediatamente el canal.

✓ DtOutpChannel::GetFlags

Obtiene el estado de las banderas del canal de salida

Tabla 2.18 DtOutpChannel::GetFlags

DTAPI_RESULT DtOutpChannel::GetFlags (int&, int&);	
[out] int& Status,	Parámetro de salida que describe el estado actual de la bandera
[out] int& Latched	Parámetro de salida que enclava el valor del estado de la bandera

✓ DtOutpChannel::SetFifoSize

Establece el tamaño de la cola de transición FIFO en el dispositivo con un valor específico (SetFifoSize), o a su valor máximo soportado por el canal (SetFifoSizeMax).

Tabla 2.19 DtOutpChannel::SetFifoSize

DTAPI_RESULT DtOutpChannel::SetFifoSize(int);	
[in] int FifoSize	El tamaño a configurar en bits. FifoSize debe ser un múltiplo de 16.

✓ DtOutpChannel::SetModControl

Establece los parámetros de control de modulación, es decir permite el manejo de distintos estándares de transmisión. Para el sistema ISDB-Tb puede ser usado para realizar multiplexación jerárquica o uso de BTS.

En el caso de utilizar BTS de 204 bytes con información TMCC SetModControl(DtIsdbtPars) es usado; caso contrario nos referimos a la siguiente información.

Tabla 2.20 DtOutpChannel::SetModControl

DTAPI_RESULT DtOutpChannel::SetModControl (DtIsdbtPars&, int, int, int);	
[in] DtIsdbtPars&	Hace referencia a la clase DtIsdbtPars y sus variables globales que previamente se han configurado
[in] int ParXtra0,	Parámetro de modulación extra #0 es la operación OR de los valores de los campos: Número Total Inicial de Segmentos, Ancho de Banda, Tasa de Muestreo y Sub-Canal. Para el presente proyecto se usa los valores: Número Total Inicial de Segmentos: DTAPI_ISDBT_SEGM_13 Ancho de Banda: DTAPI_ISDBT_BW_6MHZ Tasa de Muestreo: DTAPI_ISDBT_SRATE_1_1 Sub-Canal: No usado.
[in] int ParXtra1,	No Usado
[in] int ParXtra2	No Usado

✓ DtOutpChannel::SetTsRateBps

Establece la tasa de transmisión del canal. Se debe tomar en cuenta que la tasa del Transport Stream es basada en los 188 bytes de cada paquete. Se recomienda usar la presente configuración después de la vinculación del canal y control del modo de transmisión.

Tabla 2.21 DtOutpChannel::SetTsRateBps

DTAPI_RESULT DtOutpChannel::SetTsRateBps (int, int);	
[in] int ClockGenMode,	Define el modo de generación de reloj. En el presente proyecto no es usado
[in] int TsRate	La tasa del Transport Stream es especificada en bps.

✓ DtOutpChannel::SetTxMode

Configura el modo de transmisión de canal

Tabla 2.22 DtOutpChannel::SetTxMode

DTAPI_RESULT DtOutpChannel::SetTxMode (int, int);	
[in] int TxMode,	<p>El parámetro de modo de transmisión puede tener los siguientes valores:</p> <p>DTAPI_TXMODE_188: Se asume que los paquetes son de 188 bytes y se transmiten sin ninguna modificación.</p> <p>DTAPI_TXMODE_204: Se asume que los paquetes son de 204 bytes y se transmiten sin ninguna modificación.</p> <p>DTAPI_TXMODE_ADD16: Se asume que los paquetes son de 188 bytes y se añaden 16 bytes extras (0) a cada paquete.</p> <p>DTAPI_TXMODE_RAW: No se asume nada en la estructura del paquete y se transmite sin modificación. No se transmiten paquetes nulos.</p>
[in] int StuffMode	Controla el comportamiento de la salida cuando no existen paquetes en la cola FIFO. Si obtiene el valor de '1' (activo). Se transmiten paquetes nulos del tamaño del paquete seleccionado. Caso contrario no se transmite nada

✓ DtOutpChannel::Write

Escribe los bytes de datos en el canal de salida.

DTAPI_RESULT DtOutpChannel::Write (char*, int);	
[in] char* pBuffer,	Puntero señalando el buffer que contiene el TS a transmitir
[in] int NumBytesToWrite	Numero de bytes que se van a transmitir obtenidos del archivo TS

✓ DtOutpChannel::SetTxControl

Establece el estado del control de transmisión del canal. Permite controlar el inicio o fin del streaming de datos. Se recomienda su configuración después de la vinculación al dispositivo.

Tabla 2.23 DtOutpChannel::SetTxControl

DTAPI_RESULT DtOutpChannel::SetTxControl (int);	
[in] int TxControl	<p>Define el estado de control de acuerdo a los siguientes valores:</p> <p>DTAPI_TXCTRL_IDLE: Transmisión de paquetes desde la cola FIFO esta deshabilitada. Sin embargo se envían paquetes nulos</p> <p>DTAPI_TXCTRL_HOLD: Los datos se transfieren a la FIFO hasta llenar su capacidad. Sin embargo la transmisión de paquetes es nula como el estado anterior.</p> <p>DTAPI_TXCTRL_SEND: Opera normalmente. La transferencia de datos y transmisión de paquetes está habilitada</p>

2.3.3 Contenido de Estructuras ISDB-Tb

Una estructura es una forma de agrupar y organizar un conjunto de variables con el objetivo de simplificar su manipulación. DTAPI ofrece algunas estructuras para facilitar la programación de una aplicación. A continuación se detallan las usadas en el mencionado proyecto.

2.3.3.1 Estructura DtDeviceDesc

Es la estructura que contiene las variables que describen a un dispositivo DekTec. A través del método DtDevice::GetDescriptor se cargan valores a las variables que detallamos a continuación. En el presente proyecto se obvian algunas variables de poca importancia

Tabla 2.24 Struct DtDeviceDesc

Struct DtDeviceDesc {...};	
__int64 m_Serial;	Número serial único que identifica al dispositivo DekTec.
int m_UsbAddress;	Identifica la dirección USB del dispositivo.
int m_TypeNumber;	Variable que corresponde al número del tipo de dispositivo. En este caso será 215
int m_DeviceId;	Número de identificación del dispositivo
int m_VendorId;	Número de identificación del proveedor del dispositivo
int m_FirmwareVersion;	Número de la versión de firmware cargada en el dispositivo
int m_FirmwareVariant;	Número de la variante de firmware cargada en el dispositivo

2.3.3.2 Estructura DtIsdbtLayerPars

Es la estructura que describe los parámetros de modulación ISDB-T para una capa jerárquica. Esta estructura es usada en la clase DtIsdbtPars como un array de tres estructuras para las capas A, B y C.

Tabla 2.25 Struct DtIsdbtLayerPars

struct DtIsdbtLayerPars {...};	
int m_NumSegments;	Número que representa la cantidad de segmentos usados en la capa jerárquica.
int m_Modulation;	Tipo de modulación aplicada a la capa. Las variables pueden ser: DTAPI_ISDBT_MOD_DQPSK, DTAPI_ISDBT_MOD_QPSK, DTAPI_ISDBT_MOD_16QAM, DTAPI_ISDBT_MOD_64QAM

int m_CodeRate;	Variable que corresponde a la tasa del código convolucional aplicado a la capa DTAPI_ISDBT_RATE_1_2, DTAPI_ISDBT_RATE_2_3 DTAPI_ISDBT_RATE_3_4, DTAPI_ISDBT_RATE_5_6 DTAPI_ISDBT_RATE_7_8
int m_TimeInterleave;	Variable que representa la longitud del entrelazado de tiempo aplicado a la capa. Puede tomar los siguientes valores: 0: 0 para los modos 1, 2, 3 1: 4, 2, 1 respectivamente 2: 8, 4, 2 respectivamente 3: 16, 8, 4 respectivamente
int m_BitRate;	Tasa de bits asignada llamando el método DtIsdbtPars::ComputeRates.

CAPÍTULO III

3. DESARROLLO DEL SOFTWARE

El desarrollo de la plataforma de control para la moduladora DTU-215 consta de dos partes: el código nativo en C++ que usa la librería DTAPI y controla la tarjeta moduladora DTU-215; y la sección en Java que integra el código C++ como otra librería, presenta una interfaz de usuario amigable y realiza validación de datos. La figura 3.1 muestra los distintos niveles de abstracción del presente proyecto, siendo los dos últimos los desarrollados por el autor.

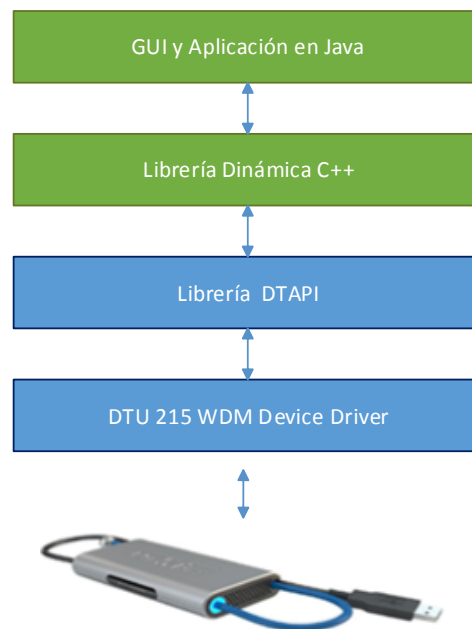


Figura 3.1 Niveles de Abstracción de la Plataforma de Control para la DTU-215

Se desarrolló una interfaz en Java para generar la parte medular de un nuevo multiplexor integrado para la operación, procesamiento y control de señales de televisión digital en tiempo real al igual que otras soluciones comerciales que utilizan los mismos equipos DekTec. La combinación de las dos partes da como resultado un programa que permite utilizar la tarjeta moduladora y permite la futura integración de software para el desarrollo de otras aplicaciones.

La plataforma de control fue desarrollada utilizando los entornos de desarrollo integrado Visual Studio y Netbeans. A continuación especificaremos estas herramientas utilizadas

3.1 ENTORNOS DE DESARROLLO INTEGRADO

3.1.1 IDE Visual Studio 10.0

Visual Studio 2010 es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistema operativo Windows. Es un programa informático compuesto por un conjunto de herramientas orientadas a desarrolladores de software que soporta multilenguaje como Python, C, C++, PHP, etc. Además es orientado al desarrollo de software en equipos de trabajo por lo que simplifica la integración de librerías, proyectos y demás. [16]

En el presente proyecto se utiliza el mencionado IDE para el desarrollo de la biblioteca de enlace dinámico (DLL por sus siglas en inglés) que será posteriormente utilizado en nuestro programa de Java. Visual Studio fue utilizado dado que DekTec desarrolló su librería DTAPI en este IDE y el manual recomienda su uso. En la siguiente sección mencionaremos características generales que sirvieron en el desarrollo del proyecto.

3.1.1.1 Características Generales

Visual Studio 2010 tiene distintas funcionalidades que permite conectar unas a otras sean tecnología Microsoft o no. La figura 3.2 indica los componentes principales y algunas otras tecnologías que son las más usadas.

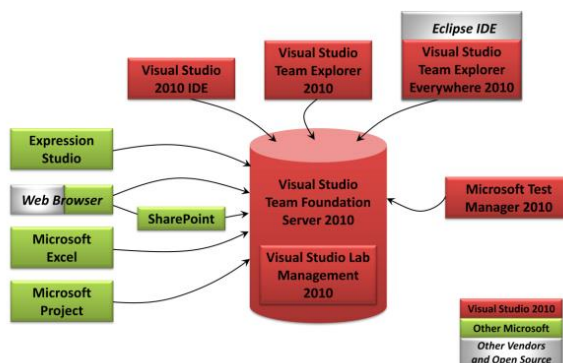


Figura 3.2 Componentes principales Visual Studio 2010 [16]

Como cualquier IDE, Visual Studio 2010 provee una interfaz gráfica moderna que incluye mecanismos sencillos para manejar código y configuración de archivos, con la habilidad de mostrar diferentes partes del código en distintos colores simplificando al desarrollador su uso. Además, soporta el uso de múltiples monitores, con diferentes partes del código mostrando en las diferentes pantallas.

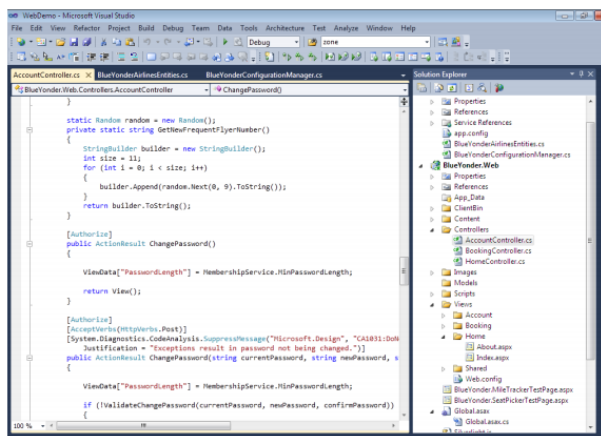


Figura 3.3 Interfaz gráfica de Visual Studio 2010

3.1.2 IDE Netbeans

Netbeans IDE es el entorno de desarrollo oficial para Java que permite desarrollar aplicaciones de escritorio, móviles y web utilizando el lenguaje de programación Java; así como aplicaciones con HTML, JavaScripts y CSS. Netbeans es un IDE modular,

normalizado, libre, escrito en lenguaje Java y que tiene una vasta comunidad de usuarios y desarrolladores alrededor del mundo.

En el desarrollo del presente proyecto fue utilizado para desarrollar la integración de la librería previamente realizada en C++ con una interfaz de usuario amigable y permitiendo integrar otras funcionalidad a la plataforma.

3.1.2.1 Características Generales

Netbeans presenta editores, analizadores de código, convertidores, etc. que permiten desarrollo y mejoramiento de aplicaciones. Además, características como mejores funcionalidad de edición JavaScript, soporte para usar estructuras Spring de soporte web y una optimización de recursos al compartir librerías entre proyectos dependientes. [17]

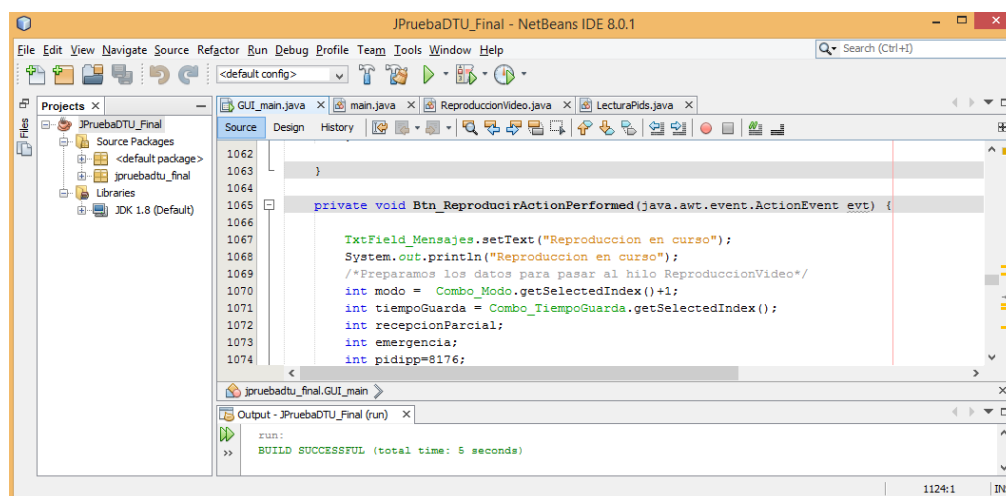


Figura 3.4 Interfaz gráfica de Netbeans

3.2 DESARROLLO APLICACIÓN

Como se mencionó anteriormente, el desarrollo de la aplicación consta de dos partes: Una biblioteca de enlace dinámico (DLL por sus siglas en inglés) desarrollada en código nativo en C++ y la sección que muestra la interfaz gráfica de usuario en Java. Para integrar la

librería creada en C++ con el programa desarrollado en Java se utilizó la herramienta Java Native Interface (JNI). A continuación mencionaremos el desarrollo del código en Java.

3.2.1 Código Java

La aplicación en Java consta de 4 clases que permiten la interacción del usuario, la validación de datos y la comunicación con la librería DLL. La figura 3.5 muestra el diagrama con las clases de la aplicación en Java. A continuación vamos a especificar las 4 clases.



Figura 3.5 Diagrama UML Aplicación Java

3.2.1.1 Clase IntegracionJNI

Contiene las variables y métodos que van a ser utilizados en la librería DLL. Esta clase únicamente declara los métodos nativos en Java, la implementación se realiza en C++. En las secciones Integración con JNI y Código C++ se especifica los métodos y pasos realizados. A continuación se enlista los métodos declarados y utilizados en esta clase.

Tabla 3.1 Métodos declarados en la Clase IntegracionJNI

IntegracionJNI
String LeerError()
boolean ConectarPuerto()
boolean LimpiarBanderas()
boolean AbrirArchivo(String ts_path)
boolean Configuracion_Tx_TodoCapaA(boolean flag_TMCC, int[] valores_config)
int[] Configuracion_Tx(boolean flag_TMCC, int[] valores_config, int[] PIDs_a, int[] PIDs_b, int[] PIDs_c, int size_a, int size_b, int size_c)
boolean InitTxParams(double frecuencia, int size_pack, int nivelDbm)
boolean ReproducirTs(int wrap);
boolean PararTs();
boolean LeerBanderas();
boolean CerrarPrograma();

3.2.1.2 Clase ReproducciónVideo

Es una clase del tipo Runnable que toma los datos de configuración previamente validados y los utiliza con los métodos de la librería C++ para realizar la configuración de los parámetros de transmisión y la reproducción del archivo TS. En esta clase se utilizan los métodos LeerError, LimpiarBanderas, Configuracion_Tx_TodoCapaA, Configuracion_Tx, InitTxParams, ReproducirTs, LeerBanderas();

3.2.1.3 Clase LecturaPids

Es una clase del tipo Runnable que permite la identificación de los PIDs de los distintos servicios embebidos en el archivo TS. Analiza el archivo a nivel de bytes, identifica los PIDs, guarda los resultados en un array de variables int e imprime en pantalla su resultado en la tabla de la interfaz gráfica.

3.2.1.4 Clase GUI_Main

Es la clase principal la cual contiene todos los objetos de la interfaz gráfica, métodos internos y utiliza objetos de las otras clases previamente mencionadas. En esta clase se realiza la interacción con el usuario, validación de datos y llama los métodos de la librería C++. La figura 3.6 detalla la interfaz gráfica desarrollada.

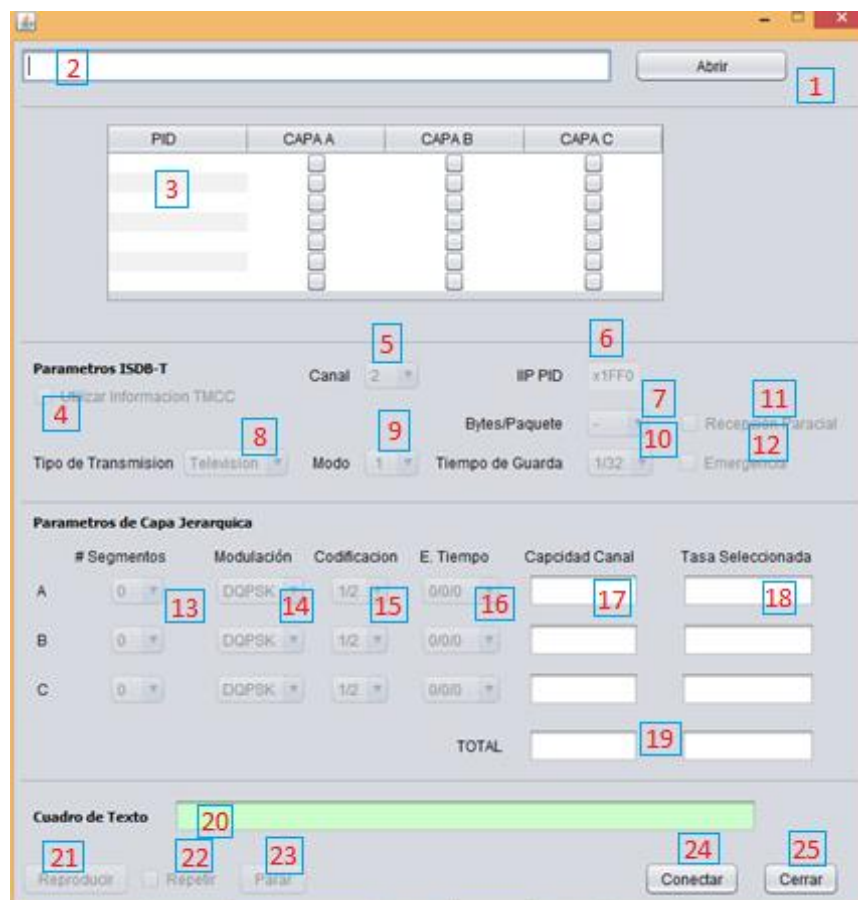


Figura 3.6 Interfaz gráfica de la plataforma de control DTU-215

Tabla 3.2 Detalle de los elementos de la interfaz gráfica de la aplicación en Java

#	Variable	Descripción	Tipo
1	Abrir	Permite escoger el archivo que se va a transmitir. Internamente se genera un evento para validar que el archivo sea un Transport Stream y el número de bytes/paquete. Si es un archivo .ts habilita los objetos gráficos e imprime en pantalla los PIDs de ser posible, caso contrario imprime un mensaje de error	Botón
2	Directorio Transport Stream	Indica el nombre y ubicación del archivo TS	Display
3	Tabla PID	Indica, de ser posible los PIDs de los servicios embebidos en el archivo TS	Display
4	Utilizar información TMCC	Habilitado únicamente cuando el archivo TS se ha validado que tiene 204 bytes/paquete. Si está seleccionado permite leer y utilizar la información TMCC para visualización del usuario y su posterior reproducción.	Check
5	Canal	Permite seleccionar el canal de televisión en el que se va a transmitir	Desplegar
6	IIP PID	Indica el PID usado para transferir la ISDB-T Information Packet (IIP) al modulador. IIP contiene la información TMCC. Por default es predefinida como 0x1FF0	Display
7	Bytes/ Paquete	Indica el número de bytes por cada paquete del archivo TS. Esta información es importante para el momento de configurar los parámetros de transmisión.	Desplegar
8	Tipo de Transmisión	Indica el tipo de transmisión. Por default es predefinida como Televisión (13 segmentos),	Display
9	Modo	Permite seleccionar el modo de transmisión, los cuales definen el número de portadoras por segmento. Estos pueden ser modo 1 (108), modo 2 (216) o modo3 (432).	Desplegable
10	Intervalo de guarda	Permite seleccionar el intervalo de guarda de la transmisión OFDM. Estos pueden ser 1/4, 1/8, 1/16, 1/32.	Desplegable

11	Recepción Parcial	Si está activo, permite el uso de transmisión one-seg y la capa A debe tener un segmento dedicado para esto	Check
12	Transmisión de Emergencia	Si está activo, se habilita la bandera de alarma de emergencia (1 bit) en la información TMCC	Check
13	Numero de Segmentos	Permite seleccionar el número de segmentos asignado a las capas jerárquicas	Desplegable
14	Modulación	Permite seleccionar el tipo de modulación para cada capa jerárquica. Las modulaciones pueden ser DQPSK, QPSK, 16QAM o 64QAM	Desplegable
15	Tasa de código interno	Permite seleccionar la tasa de código interno para cada capa jerárquica. Las tasas pueden ser 1/2, 2/3, 3/4, 5/6, 7/8.	Desplegable
16	Tiempo de entrelazado	Permite seleccionar el tiempo de entrelazado. Este permite tener una señal más robusta y depende del modo de transmisión escogido. El tiempo puede ser 0, 4, 8, 16 (modo 1); 0, 2, 4, 8 (modo2); 0, 1, 2, 4 (modo 3).	Desplegable
17	Tasa de transmisión (bps)	Este campo muestra la tasa de transmisión permitida en cada capa calculado en base a las configuraciones previas	Display
18	Tasa de Transmisión Seleccionada (bps)	Este campo muestra la tasa de transmisión seleccionada de los distintos programas o servicios que han sido asignados a la capa	Display
19	Total	Indica el total de segmentos asignados a las capas jerárquicas. El valor debe ser 13.	Display
20	Cuadro de Mensajes	Indica si la configuración es válida y muestra cualquier error	Display
21	Reproducir	Controla el botón de Reproducir la transmisión del archivo TS.	Botón
22	Repetir	Cuando está activado, el archivo TS se reproduce infinitamente	Check
23	Parar	Controla el botón de Parar la transmisión del archivo TS	Botón

24	Conectar	Permite entablar comunicación entre la aplicación y la tarjeta moduladora DTU-215	Botón
25	Cerrar	Corta la comunicación con la tarjeta y cierra la aplicación	Botón

3.2.2 Integración Java/C++

JNI especifica el protocolo de comunicación entre el código Java y un código externo nativo escrito en otro lenguaje permitiendo interactuar entre ellos.

JNI permite a los desarrolladores de software utilizar herramientas fuera del alcance de la Máquina Virtual de Java, utilizar herramientas ya desarrolladas en otros lenguajes sin la necesidad de reescribirlo en Java y mejorar el rendimiento de ciertos procesos del programa utilizando código nativo como C o Assembler.



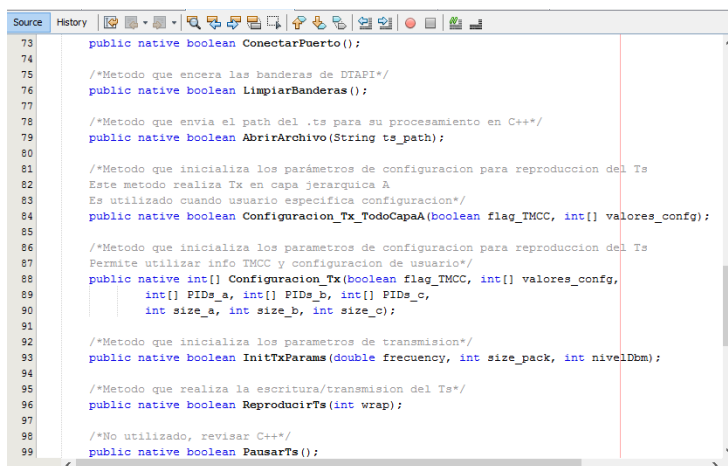
Figura 3.7 Integración Java y C++ con JNI

En nuestro caso, para poder utilizar las clases, estructuras y métodos de la librería DTAPI para controlar a la tarjeta moduladora implementamos una biblioteca de enlace dinámico que a través de JNI interactúe con el código en Java. Se debe tomar en cuenta en bajos niveles de abstracción, ciertas variables como cadenas de caracteres, objetos, etc. se tratan de forma distinta en código Java y código C++. Por lo tanto en la implementación del proyecto se limitó a lo imprescindible la interacción entre los distintos códigos

3.2.2.1 Implementación JNI

Para realizar la implementación de JNI se realizó el siguiente proceso:

- ✓ Crear la clase IntegracionJNI.java mencionada previamente donde declaramos los métodos nativos que vamos a implementar en C++.



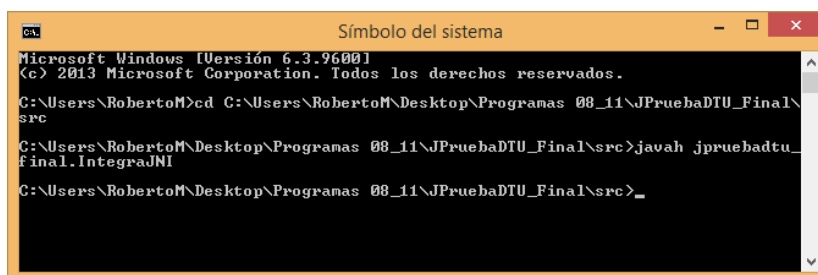
```

73 public native boolean ConectarPuerto();
74
75 /*Metodo que encera las banderas de DTAPI*/
76 public native boolean LimpiarBanderas();
77
78 /*Metodo que envia el path del .ts para su procesamiento en C++*/
79 public native boolean AbrirArchivo(String ts_path);
80
81 /*Metodo que inicializa los parámetros de configuracion para reproduccion del Ts
82 Este metodo realiza Tx en capa jerarquica A
83 Es utilizado cuando usuario especifica configuracion*/
84 public native boolean Configuracion_Tx_TodoCapaA(boolean flag_TMCC, int[] valores_config);
85
86 /*Metodo que inicializa los parametros de configuracion para reproduccion del Ts
87 Permite utilizar info TMCC y configuracion de usuario*/
88 public native int[] Configuracion_Tx(boolean flag_TMCC, int[] valores_config,
89 int[] FIDs_a, int[] FIDs_b, int[] FIDs_c,
90 int size_a, int size_b, int size_c);
91
92 /*Metodo que inicializa los parametros de transmision*/
93 public native boolean InitTxParams(double frecuency, int size_pack, int nivelDbm);
94
95 /*Metodo que realiza la escritura/transmision del Ts*/
96 public native boolean ReproducirTs(int wrap);
97
98 /*No utilizado, revisar C++*/
99 public native boolean PausarTs();

```

Figura 3.8 Declaración de métodos nativos en Java

- ✓ Compilar la clase IntegracionJNI.java resultando en el archivo tipo class IntegracionJNI.java
- ✓ Utilizando el ejecutable símbolos del sistema (cmd.exe) y el comando *javah* generamos el archivo de cabecera IntegraciónJNI.h que contiene el prototipo de función para la implementación de los métodos nativos. La herramienta javah es parte del Java Development Kit (JDK).



```

Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\RobertoM>cd C:\Users\RobertoM\Desktop\Programas 08_11\JPruebaDTU_Final\src
C:\Users\RobertoM\Desktop\Programas 08_11\JPruebaDTU_Final\src>javah jpruebadtu_final.IntegracionJNI
C:\Users\RobertoM\Desktop\Programas 08_11\JPruebaDTU_Final\src>_

```

Figura 3.9 Generación de cabecera utilizando *javah*

- ✓ El archivo IntegracionJNI.h integramos al proyecto en Visual Studio y la declaración de los métodos copiamos al archivo CLibDTUFinal.cpp donde implementaremos los mencionados métodos.

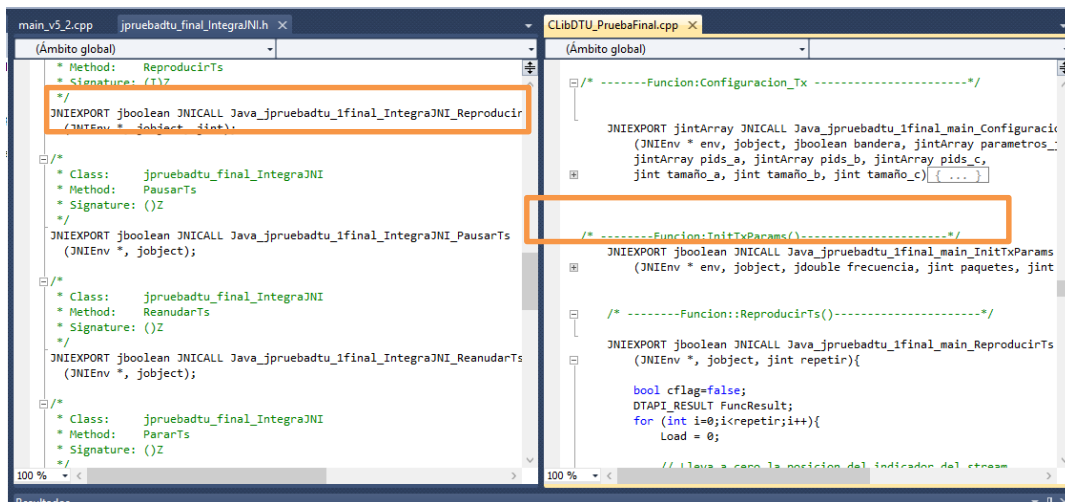


Figura 3.10 Utilización de InegracionJNI.h para implementar los métodos nativos

- ✓ Al proyecto en Visual Studio configuramos como Biblioteca dinámica, en el campo Propiedades>Propiedades de Configuración>General>Tipo de Configuración>Biblioteca dinámica (dll).

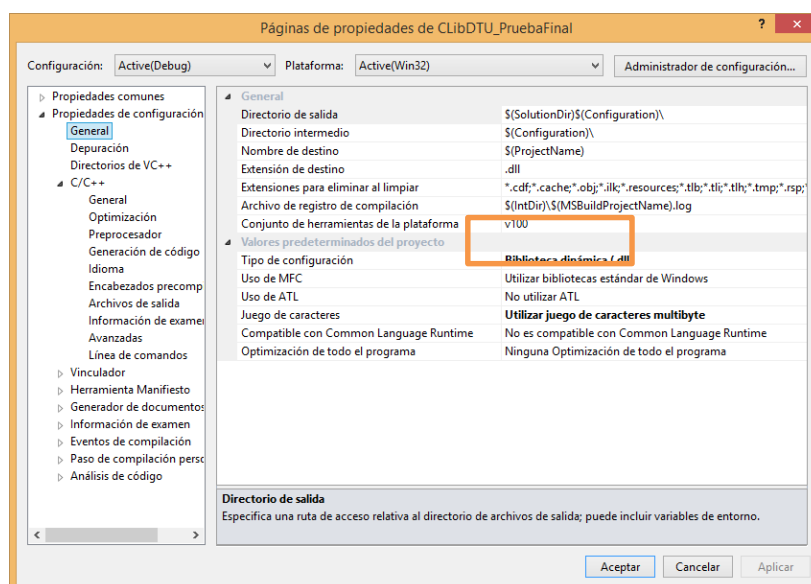


Figura 3.11 Configuración proyecto C++ como Librería Dinámica

- ✓ Añadimos en los directorios de inclusión adicionales, las carpetas *include* y *win32* del JDK de Java. Además la carpeta donde se encuentre el archivo de cabecera *InegraJNI.h* previamente generado

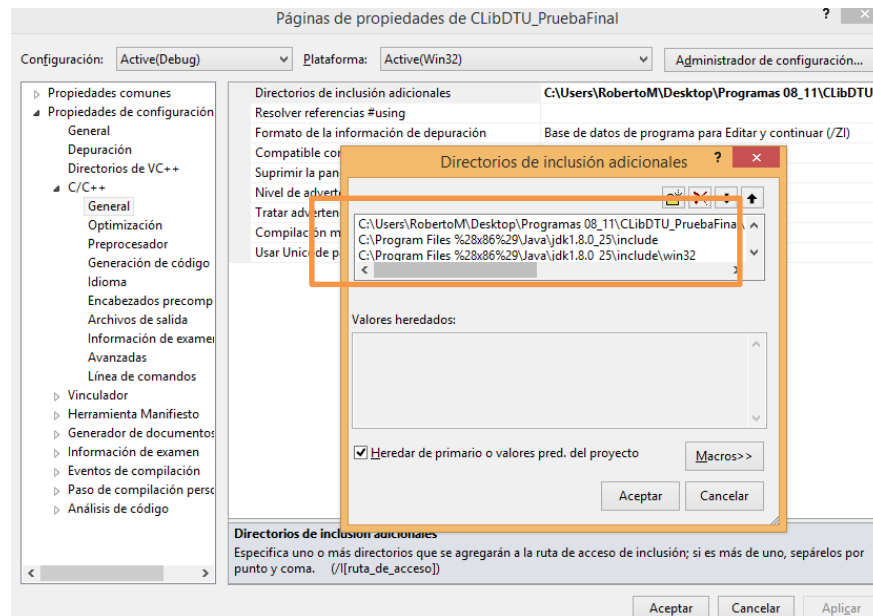


Figura 3.12 Inclusión de directorios adicionales en proyecto C++

- ✓ Procedemos a implementar el código C++ de los métodos que vamos a desarrollar. Esta sección es detallada posteriormente
- ✓ Una vez que hemos desarrollado los métodos y no tenemos errores de sintaxis u otros, compilamos el proyecto el cual da como resultado el archivo *CLibDTUFinal.dll*

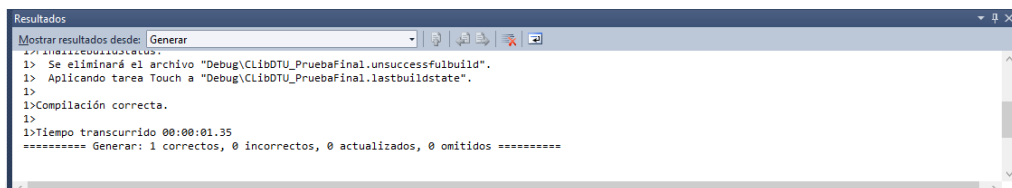


Figura 3.13 Compilación proyecto C++

- ✓ De vuelta en Netbeans añadimos el directorio de la ubicación de la librería dinámica en las propiedades del proyecto en Java. Propiedades>Run>Virtual Machine Options utilizando la sentencia `-Djava.library.path="Directorio"`

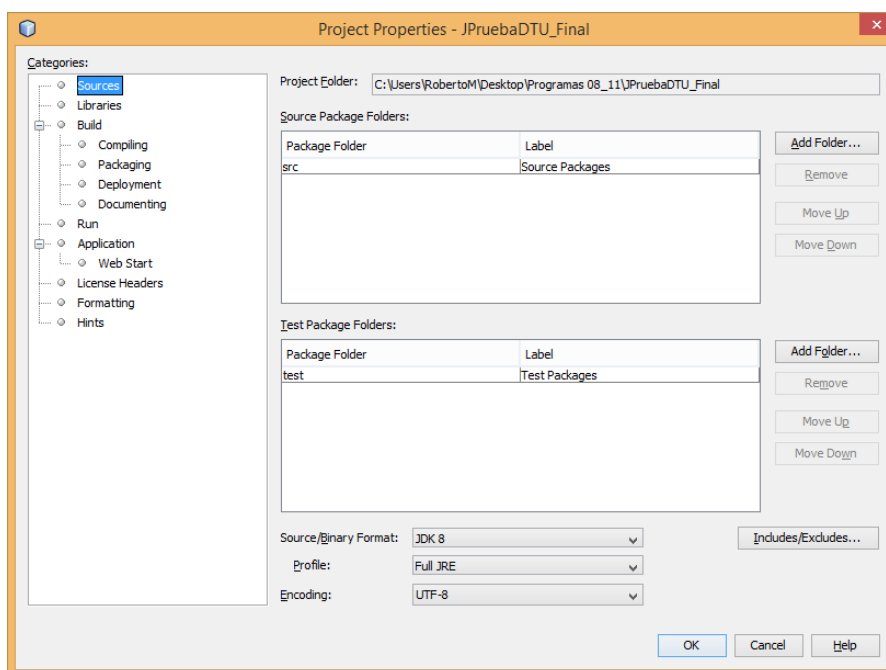


Figura 3.14 Inclusión directorio ubicación Liberia Dinámica

- ✓ Corremos el proyecto en Java.

3.2.3 Código C++

El desarrollo del software que gestiona el control de la tarjeta moduladora fue diseñado aprovechando al máximo los recursos que permite la librería DTAPI. Como se especificó en una sección previa, DTAPI permite ser funcional para otros modelos de productos DekTec y soporta diferentes estándares de transmisión distintos al Japonés-Brasilero. Nuestra aplicación fue desarrollada para utilizar la tarjeta moduladora DTU-215 y soportar el estándar de televisión ISDB-Tb.

Los pasos que seguimos configurando el proyecto en Visual Studio se mencionó en la sección anterior. A continuación explicaremos la implementación de los métodos en la librería dinámica.

La Librería DLL consta de 14 métodos que realizan la conexión, configuración, transmisión, entre otras funcionalidades permitiendo reproducir archivos Transport Stream. A continuación especificaremos los métodos desarrollados. Es importante mencionar que en la explicación de las variables de los métodos no se mencionaran JNIEnv *env y jobject ya que las mismas se generan automáticamente y permiten el uso de JNI en el código nativo.

✓ String LeerError()

En caso de existir, permite leer el tipo de error producido al realizar una acción con la DTAPI y la tarjeta moduladora. C++ retorna una variable del tipo String donde se encuentra el mensaje de error previamente producido. Dependiendo del error producido, el mensaje puede ser cadenas de caracteres constantes propias del proyecto previamente definidos y o cadena de caracteres generados por la librería DTAPI. Ver Anexo 1.

Tabla 3.3 Método LeerError

JNIEXPORT jstring JNICALL Java_jpruebadtu_1final_main_LeerError (JNIEnv *env, jobject)	
[out] jstring,	Variable con el texto del error generado

✓ boolean AdjuntarTarjeta()

Permite inicializar la conexión entre la tarjeta moduladora y la PC. El presente método únicamente permite conectarse con la moduladora DekTec DTU-215. Ver Anexo 1.

Tabla 3.4 Método AdjuntarTarjeta

JNIEXPORT jboolean JNICALL Java_jpruebadtu_1final_main_AdjuntarTarjeta (JNIEnv *, jobject)	
[out] jboolean,	Bandera que indica si la tarjeta moduladora se conectó satisfactoriamente

✓ boolean ConectarPuerto()

Permite conectar lógicamente con el puerto de transmisión de la tarjeta moduladora. Ver Anexo 1.

Tabla 3.5 Método ConectarPuerto

JNIEXPORT jboolean JNICALL Java_jpruebadtu_1final_main_ConectarPuerto (JNIEnv *, jobject)	
[out] jboolean,	Bandera que indica si el puerto lógico de la tarjeta moduladora se conectó satisfactoriamente

✓ boolean LimpiarBanderas()

Encera las banderas internas de control de error de la tarjeta moduladora.

Tabla 3.6 Método LimpiarBanderas

JNIEXPORT jboolean JNICALL Java_jpruebadtu_1final_main_LimpiarBanderas (JNIEnv *, jobject)	
[out] jboolean,	Bandera que indica si las banderas se enceraron satisfactoriamente

✓ boolean Configuracion_Tx_TodoCapaA(boolean flag_TMCC, int[] valores_confg)

Para realizar la configuración de los parámetros de transmisión del archivo TS, se envía desde Java a C++ una cadena de variables *int* la cual contiene la información necesaria para la configuración y transmitirla a la DTAPI. La cadena de caracteres tiene un significado de acuerdo su posición. En la tabla 3.8 se detalla la posición, significado y valores validos que pueden tomar para la transmisión.

Tabla 3.7 Detalle cadena de variable *int* para configurar parámetros de transmisión

POSICION	DESCRIPCION	VALOR	SIGNIFICADO
[0]	Modo de Transmisión	1	Modo 1
		2	Modo 2
		3	Modo 3
[1]	Tiempo de Guarda	0	1/32
		1	1/16
		2	1/8
		3	1/4
[2]	Recepción Parcial	0	Habilitada
		1	Deshabilitada
[3]	Bit de Emergencia	0	Habilitado
		1	Deshabilitado
[4]	Valor PID IIP	0 - 0xFFFF	
[5]	Numero de Segmentos en la capa A	0-13	
[6]	Numero de Segmentos en la capa B	0-13	
[7]	Numero de Segmentos en la capa C	0-13	
[8]	Codificación Interna Capa A	0	1/2
		1	2/3
		2	3/4
		3	5/6
		4	7/8
[9]	Codificación Interna Capa B	0	1/2
		1	2/3
		2	3/4
		3	5/6
		4	7/8
[10]	Codificación Interna Capa C	0	1/2

		1	2/3
		2	3/4
		3	5/6
		4	7/8
[11]	Entrelazado de Tiempo en Capa A	0	0 Para modos 1,2,3
		1	4,2,1 respectivamente
		2	8,4,2 respectivamente
		3	16,8,4 respectivamente
[12]	Entrelazado de Tiempo en Capa B	0	0 Para modos 1,2,3
		1	4,2,1 respectivamente
		2	8,4,2 respectivamente
		3	16,8,4 respectivamente
[13]	Entrelazado de Tiempo en Capa C	0	0 Para modos 1,2,3
		1	4,2,1 respectivamente
		2	8,4,2 respectivamente
		3	16,8,4 respectivamente
[14]	Modulación en la Capa A	0	DQPSK
		1	QPSK
		2	16 QAM
		3	64 QAM
[15]	Modulación en la Capa B	0	DQPSK
		1	QPSK
		2	16 QAM
		3	64 QAM
[16]	Modulación en la Capa C	0	DQPSK
		1	QPSK
		2	16 QAM
		3	64 QAM
[17]	Capacidad de Canal Capa A	0 - 23234999	
[18]	Capacidad de Canal Capa B	0 - 23234999	
[19]	Capacidad de Canal Capa C	0 - 23234999	
[20]	Bit de Llenado	1	Único valor válido de configuración

Cuando los valores se toman en la librería nativa, se los cargan en las variables de la clase y estructura para luego inicializarlos. En el presente método únicamente se puede cargar los valores en la capa jerárquica A cuando el usuario realiza su propia configuración. Ver Anexo 1.

Tabla 3.8 Método ConfiguraciónTxCapaA

JNIEXPORT	jboolean	JNICALL
Java_jpruebadtu_1final_main_Configuracion_1Tx_1TodoCapaA (JNIEnv *env, jobject, jboolean bandera, jintArray parametros_java)		
[in] jboolean bandera,	Bandera que indica si se utilizará la información embebida en el campo TMCC o no.	
[in] jintArray parametros_java,	Array de datos que tienen un valor predeterminado para la configuración de las capas jerárquicas y parámetros ISDB-Tb	
[out] jboolean	Bandera que indica si la configuración se realizó satisfactoriamente.	

✓ boolean AbrirArchivo(String ts_path)

Se toma de la variable *string* enviada desde Java el directorio y nombre de archivo a reproducirse. Si es exitoso el proceso, el puntero *stream* apunta a la dirección del puntero del archivo abierto. Ver Anexo 1.

Tabla 3.9 Método AbrirArchivo

JNIEXPORT	jboolean	JNICALL	Java_jpruebadtu_1final_main_AbrirArchivo
(JNIEnv *env, jobject object, jstring string)			
[in] jstring string,	Cadena de caracteres que contiene el directorio y nombre del archivo TS a reproducirse		
[out] jboolean,	Bandera que indica si el archivo TS se leyó satisfactoriamente		

- ✓ `int[] Configuracion_Tx(boolean flag_TMCC, int[] valores_config, int[] PIDs_a, int[] PIDs_b, int[] PIDs_c, int size_a, int size_b, int size_c)`

El presente método permite realizar la configuración de los parámetros de transmisión. Dependiendo del valor de la variable booleana *flag_TMCC* se puede utilizar la información de transmisión contenida en el campo TMCC o los parámetros de configuración realizada por el usuario. Los parámetros enviados desde Java deben pasar por una etapa previa de integración a C++. Como se menciona antes la forma en que las variables se tratan en código Java y C++ es distinta. Ver Anexo 1.

Tabla 3.10 Método ConfiguracionTx

JNIEXPORT jintArray JNICALL Java_jpruebadtu_1final_main_Configuracion_1Tx (JNIEnv * env, jobject, jboolean bandera, jintArray parametros_java, jintArray pids_a, jintArray pids_b, jintArray pids_c, jint tamaño_a, jint tamaño_b, jint tamaño_c)	
[in] jboolean bandera,	Bandera que indica si se utilizará la información embebida en el campo TMCC o no.
[in] jintArray parametros_java,	Array de datos que tienen un valor predeterminado para la configuración de las capas jerárquicas y parámetros ISDB-Tb
[in] jintArray pids_a	Array de datos que contiene el valor de los PIDs que se transmitirán en la capa jerárquica A
[in] jintArray pids_b	Array de datos que contiene el valor de los PIDs que se transmitirán en la capa jerárquica B
[in] jintArray pids_c	Array de datos que contiene el valor de los PIDs que se transmitirán en la capa jerárquica C
[in] jint tamaño_a	Numero de servicios PIDs que se transmitirán en la capar jerárquica A
[in] jint tamaño_a	Numero de servicios PIDs que se transmitirán en la capar jerárquica B
[in] jint tamaño_a	Numero de servicios PIDs que se transmitirán en la capar jerárquica C
[out] jintArray	Array de datos que devuelve la configuración de las capas jerárquicas y parámetros ISDB-Tb realizados. Es utilizada para validación de datos en Java.

Cuando vamos a tomar la información TMCC tomamos una parte del archivo TS, con la función *fread*, la almacenamos en una memoria temporal donde a nivel de bytes buscamos el campo de sincronización de los paquetes TS. Si 5 veces, cada 204 bytes está el valor 0x47 significa que en existe dicho campo. Ubicamos el indicador al inicio del primer paquete TS encontrado y utilizando la DTAPI utilizamos la información embebida en el campo TMCC. Ver Anexo 1.

Cuando vamos a transmitir con la configuración de parámetros descrita por el usuario, utilizamos la cadena de variables *jint paramteros_conf* y de acuerdo a la información almacenada en cada posición vamos cargándola en las distintas variables de la clase y estructura diseñada para ISDB-Tb de la librería DTAPI. Ver Anexo 1.

Previo a la finalización de este método, liberamos los espacios de memoria utilizados por las variables enviadas desde Java. Este es importante ya que caso contrario podría existir un desbordamiento de buffer. Ver Anexo 1.

✓ boolean InitTxParams(double frequency, int size_pack, int nivelDbm)

El presente método es el encargado de realizar la configuración de la frecuencia central de transmisión, definir para la moduladora si el TS que se va a transmitir ya contiene 204 bytes por paquete o si deberá sumar los restantes a los paquetes de 188 bytes. Vale recordar que la DTAPI agrega, los 16 bytes finales de ceros si es de requerirse. Ver anexo 1.

Tabla 3.11 Método InitTxParams

JNIEXPORT jboolean JNICALL Java_jpruebadtu_1final_main_ InitTxParams (JNIEnv * env, jobject, jdouble frequency, jint size_pack, jint nivelDbm)	
[in] jdouble frequency,	Valor en Hz de la frecuencia a la que se transmitirá el archivo TS
[in] jint size_pack,	Valor entero que indica la cantidad de bytes que tiene un paquete TS (188 o 204)
[out] jboolean	Bandera que indica si los parámetros previamente descritos se han cargado satisfactoriamente.

✓ boolean ReproducirTs(int wrap);

Este método realiza la transmisión del Transport Stream con las configuraciones previamente realizadas. El único parámetro de Java enviado es un indicador, que de requerirse, permite repetir la transmisión embebiendo los pasos en un lazo que tiende al infinito.

La primera parte del código almacena una carga inicial en la memoria de transmisión FIFO antes de transmitir sobre el estándar ISDB-Tb. Con la función *fread* tomamos segmentos de 65536 bytes del archivo TS y utilizando la DTAPI, traspasamos a la memoria FIFO. Una vez que tenemos 1048576 bytes del archivo en la memoria FIFO, comenzamos la transmisión del Transport Stream. Se ingresa en un lazo hasta que se haya tomado todos los bytes del archivo. Ver anexo 1.

Tabla 3.12 Método ReproducirTx

JNIEXPORT jboolean JNICALL Java_jpruebadtu_1final_main_ ReproducirTx (JNIEnv * env, jobject, jint repetir)	
[in] jint repetir,	Toma el valor máximo permitido para un entero y representa el número de veces que se transmitirá el archivo TS.
[out] jboolean	Bandera que indica si se reprodujo satisfactoriamente el archivo TS.

En caso de repetirse la transmisión del mismo video, se vuelve a realizar los pasos previamente mencionados. Una vez terminado el proceso se retorna un mensaje de finalización Ver anexo 1.

✓ boolean PararTs();

Permite terminar la reproducción del Transport Stream anticipadamente. Los datos que ya se encuentran en la memoria FIFO son liberados para evitar errores en otras reproducciones.

Tabla 3.13 Método PararTs

JNIEXPORT jboolean JNICALL Java_jpruebadtu_1final_main_PararTs (JNIEnv *, jobject)	
[out] jboolean	Bandera que indica si se detuvo la reproducción del archivo TS satisfactoriamente.

✓ boolean LeerBanderas();

Como se mencionó previamente, existen banderas internas de la DTAPI que son activadas cuando existen errores como error de configuración, error de transmisión, memoria FIFO vacía, etc. Este método realiza la lectura de dichas banderas.

Tabla 3.14 Método LeerBanderas

JNIEXPORT jboolean JNICALL Java_jpruebadtu_1final_main_LeerBanderas (JNIEnv *, jobject)	
[out] jboolean	Bandera que indica si se leyó las banderas de la tarjeta DTU-215 satisfactoriamente.

✓ boolean CerrarPrograma();

Realiza las desconexiones lógicas, limpiezas de banderas, buffer y memoria FIFO para terminar la conexión con la tarjeta DTU-215. Con este método se evita futuros errores al correr la aplicación nuevamente.

Tabla 3.15 Método CerrarPrograma

JNIEXPORT jboolean JNICALL Java_jpruebadtu_1final_main_CerrarPrograma (JNIEnv *, jobject)	
[out] jboolean	Bandera que indica si se terminó la conexión con la tarjeta DTU-215 satisfactoriamente.

✓ `int ValidarArchivoTS();`

Permite validar el archivo seleccionado para su reproducción. Como se mencionó previamente los paquetes de un archivo TS tienen en su cabecera el campo de sincronización (0x47). Este método busca dicho campo cada 188 y 204 bytes. En caso de repetirse 5 veces se asume que es un archivo TS válido y devuelve un valor entero para representar si sus paquetes son de 188 o 204 bytes. En caso de no encontrarse el campo de sincronización se toma como un archivo no válido

Tabla 3.16 Método ValidarArchivoTS

JNIEXPORT jint JNICALL Java_jpruebadtu_1final_main_ValidarArchivoTS (JNIEnv *, jobject)	
[out] jint	Entero que indica si el archivo TS que se va a reproducir es válido y si contiene 188 o 204 bytes por paquete.

CAPÍTULO IV

4. PRUEBAS DE TRANSMISIÓN Y RESULTADOS

Una vez desarrollada la plataforma de control para la tarjeta DTU-215, se realizaron pruebas de funcionamiento y analizamos sus resultados. A continuación se describe las pruebas de transmisión y recepción realizadas utilizando el escenario que se observa en la figura 4.1. El escenario consta de una PC Dell Core I5 1,7 GHz con 4 GB de RAM donde se desarrolló la aplicación en Java y la librería en C++. Esta se conecta al cable USB de la tarjeta DekTec DTU-215; en el puerto de salida de la tarjeta moduladora se pasa por una zona de amplificación de la señal y a continuación un divisor que conecta al Analizador de Televisión Digital Agilent CXA y a un televisor Sony que contiene el decodificador incluido que soporta el estándar ISDB-Tb.

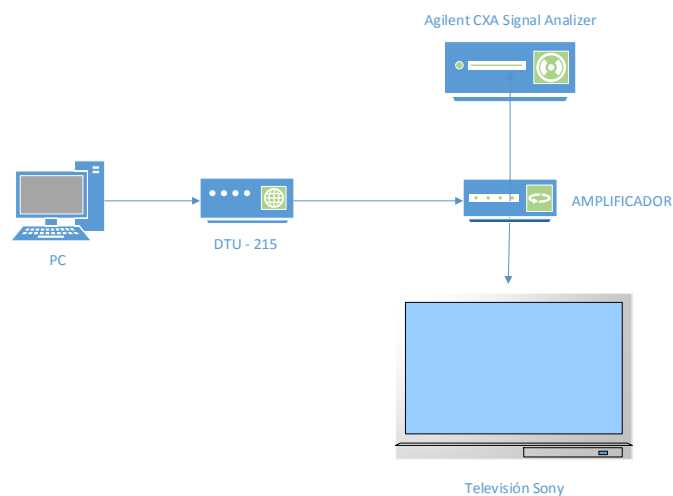


Figura 4.1 Escenario de Pruebas

Cabe mencionar que éste escenario de pruebas se realizó en las instalaciones del Laboratorio de Televisión Digital de la Universidad de las Fuerzas Armadas ESPE siendo la institución dueña de los equipos.



Figura 4.2 Escenario de pruebas de transmisión y recepción de la señal ISDB-Tb

4.1 PRUEBAS Y ANÁLISIS DE RESULTADOS PARA ARCHIVOS BROADCAST TRANSPORT STREAM

Ejecutamos la aplicación y utilizando la interfaz gráfica de usuario, se cargó un archivo BTS el cual es analizado por la aplicación. Se verifica que se ha analizado observando que el campo *Bytes/Paquete* ha cambiado y muestra el valor de 204; adicional la pestaña *Utilizar Información TMCC* ha cambiado y está habilitada. Se configuró para utilizar la información TMCC embebida en el mismo, automáticamente se visualiza la configuración ISDB-Tb y parámetros en cada capa jerárquica de la aplicación. La tabla 4.1 muestra la información que de parámetros ISDB-Tb y capa jerárquica contenida en el campo TMCC de este archivo.

Tabla 4.1 Configuración ISBD-T para pruebas de laboratorio

Intervalo de Guarda	Modo de Operación	Bytes/Paquete	Frecuencia de Transmisión
1/8	Modo 3	204	575,143 MHz

Capa	Número de Segmentos	Tasa de codificación	Modulación	Tasa Transmisión Soporta Canal
A	1	5/6	QPSK	0,5201 Mbps
B	12	3/4	16 QAM	11,234 Mbps

La figura 4.3 es una captura de pantalla de la interfaz gráfica de la aplicación sobre Java con la configuración previamente mencionada y lista para su reproducción.

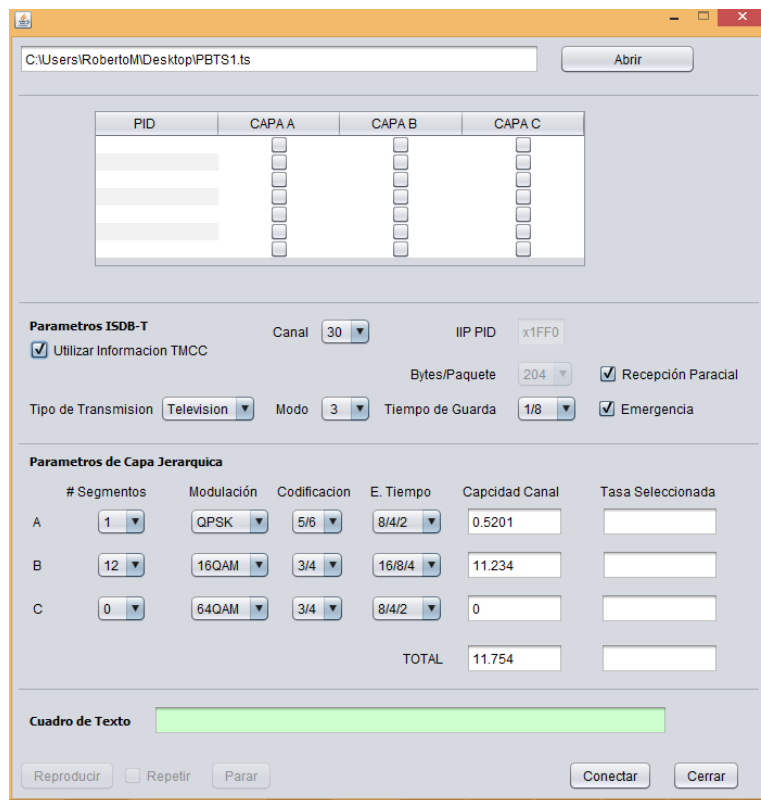


Figura 4.3 Configuración utilizada para primera prueba de transmisión

Se procedió sintonizar el canal asignado en la televisión y a reproducir el archivo TS. Aproximadamente en 4 segundos de presionar el botón Reproducir se observa en el televisor las imágenes del Transport Stream validando que se cumple la normativa que señala el estándar ISDB-Tb. La figura 4.4 muestra al televisor durante la reproducción del video y audio en la presente prueba. El tiempo de retardo entre presionar el botón Reproducir y

Durante las primeras pruebas realizadas en el analizador se verifica que la señal de transmisión es en efecto la frecuencia central antes mencionada y que el ancho de banda del canal es 6 MHz como indica el estándar ISDB-Tb. Validamos dicha información en la captura de pantalla del analizador que se muestra en la figura 4.6. Allí se observa que existe una ligera desviación a la derecha del eje indicador que muestra el analizador, esto se debe a que la frecuencia exacta de transmisión es de 575,143 MHz y la frecuencia del eje es 575 MHz.

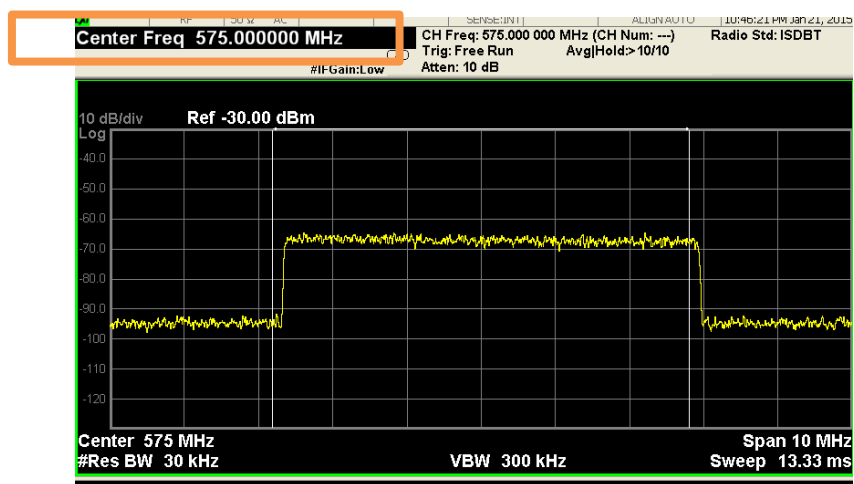


Figura 4.6 Verificación Frecuencia y Ancho de Banda de Transmisión

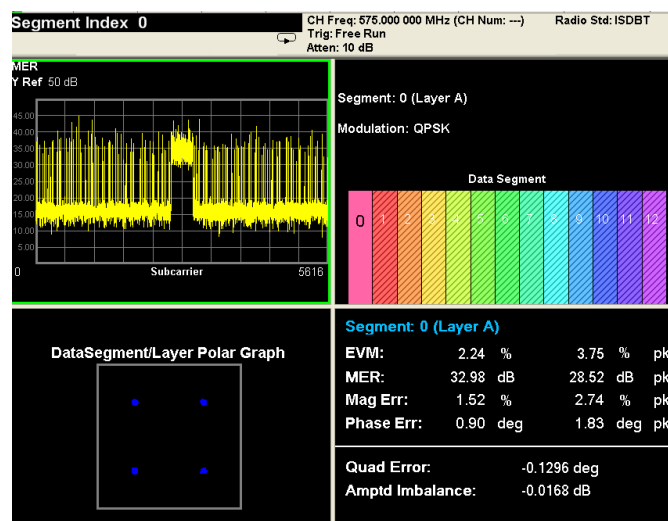


Figura 4.7 Captura parámetros de Tx capa jerárquica A

Además se observa la configuración de cada capa jerárquica que se ha configurado para estas pruebas de laboratorio. Primero verificamos que se cumpla los parámetros establecidos para la capa jerárquica A. En la figura 4-7 nos percatamos que el segmento 0 ha sido asignado para dicha capa y que QPSK es la modulación de esta sub portadora. También se muestra la constelación de dicha modulación y los valores de tasa de error binario y de magnitud del vector de error. Es importante recordar que el segmento 0 se encuentra ubicado en la frecuencia central del canal. Por lo tanto la imagen que muestra el analizador no representa la ubicación en frecuencia de dicha subportadora.

Para la capa jerárquica B la figura 4.8 muestra que desde el segmento 1 se ha asignado a dicha capa. Además observamos que la modulación es 16QAM como se definió en la configuración de la tabla 4.1.

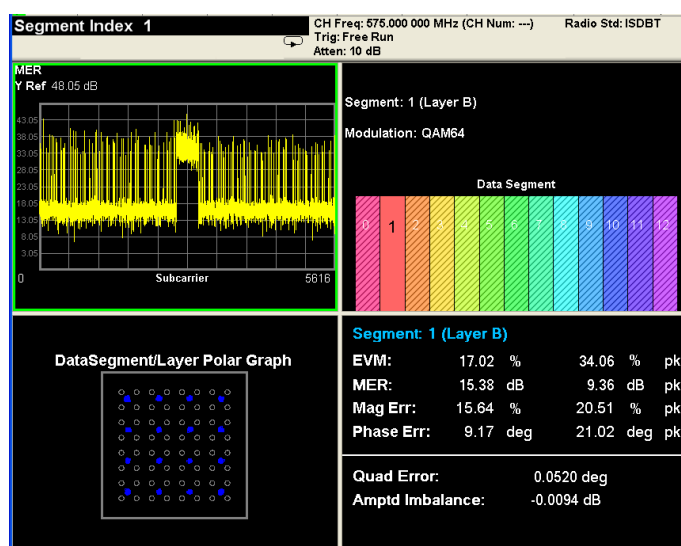


Figura 4.8 Captura parámetros de Tx capa jerárquica B

Para finalizar esta primera prueba constatamos que la configuración utilizada en efecto sea la del campo TMCC. El analizador nos permite decodificar este campo de la trama y presentar en su pantalla. La figura 4.9 valida dicha información mostrando que la configuración que muestra la aplicación es la información embebida TMCC.

TMCC Decoding Results				
		Current	Next	Current Settings
Partial Reception		Yes	Yes	Yes
Layer A:	Modulation Scheme	QPSK	Unused layer	QPSK
	Code Rate	5/6	Unused layer	
	Interleaving Length	2	Unused layer	
	Segs	1 Segment	Unused layer	1 Segment
Layer B:	Modulation Scheme	16QAM	Unused layer	64QAM
	Code Rate	3/4	Unused layer	
	Interleaving Length	4	Unused layer	
	Segs	12 Segment	Unused layer	12 Segment
Layer C:	Modulation Scheme	Unused layer	Unused layer	Unused layer
	Code Rate	Unused layer	Unused layer	
	Interleaving Length	Unused layer	Unused layer	
	Segs	Unused layer	Unused layer	Unused layer
System Descriptor			ISDB-T	
Indicator of Transmission-parameter Switching			Normal value	
Startup Control			ON	
Phase Correction			0 (No Phase Correction)	

Figura 4.9 Decodificación Información TMCC

4.2 PRUEBAS Y ANÁLISIS DE RESULTADOS PARA ARCHIVOS TRANSPORT STREAM

Como prueba adicional cargamos un archivo TS que no contenga información TMCC. En este caso nuestra aplicación además de mostrar en el campo *Bytes/Paquete* el valor 188 y deshabilitar el campo *Utilizar la Información TMCC*, captura los valores PID de los servicios del programa. Dichos PIDs observamos en la tabla superior de la figura 4.10 y nos permitirá seleccionar en cual capa jerárquica se transmitirá dicho servicio.

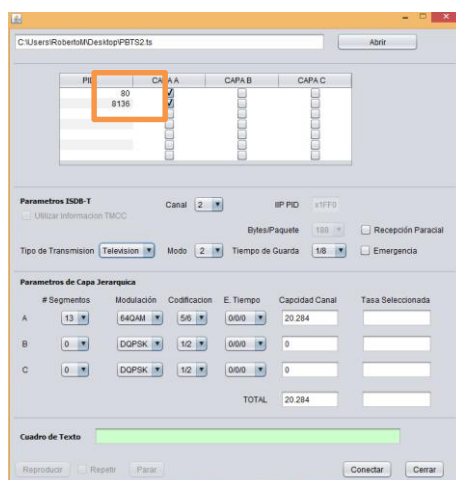


Figura 4.10 Configuración utilizada para segunda prueba de transmisión

Como en el presente caso no contamos con información TMCC, el usuario de la aplicación debe especificar los parámetros de transmisión que se van a utilizar. Durante la presente prueba se escogió transmitir en el canal 30 con modo de transmisión 2 y tiempo de guarda 1/8. Adicional la configuración de las capas jerárquicas se decidió utilizar los 13 segmentos para la capa A con una modulación de capa 64-QAM, codificación interna 5/6 y entrelazado de tiempo 0. Observamos que el campo *Capacidad de Canal* calculó la misma utilizando la ecuación descrita en el capítulo 1 e implementada en la aplicación Java. El valor calculado es 20,284 Mbps. Igual que en el caso anterior toma un tiempo aproximado de 4 segundos en observar la señal transmitida por la televisión y verificar que se cumplan los parámetros de transmisión.



Figura 4.11 Captura del Televisor Sony durante la recepción de la segunda señal de prueba

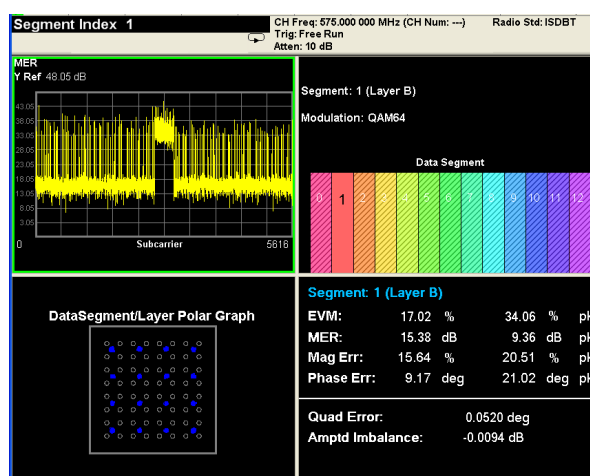


Figura 4.12 Captura parámetros de Tx capa jerárquica A

Adicional observamos la configuración de uno de los trece segmentos que han sido asignados en la capa A. Observando la figura 4.12 verificamos que se cumple la modulación configurada en dicha subportadora .

4.3 COMPARACIÓN ENTRE LA PLATAFORMA DE CONTROL DESARROLLADA Y STREAM XPRESS

Dado que el presente proyecto se basó en el software propietario Stream Xpress, realizar una comparación entre estos es esencial. Hemos separado en dos categorías la comparación de las aplicaciones. A continuación explicamos las mismas.

4.3.1 Interfaz Gráfica de Usuario

Observamos que la interfaz gráfica de usuario generada es amigable y similar a la de la aplicación Stream Xpress. Sin embargo, no cuenta con servicios como información interna de los archivos o gráfico a tiempo real de la tasa de transmisión sobre el canal. Las mencionadas diferencias no se las implementó dado que se necesita un análisis profundo a nivel de paquetes de los archivos TS a reproducirse; y dicho análisis está fuera del alcance del presente proyecto.

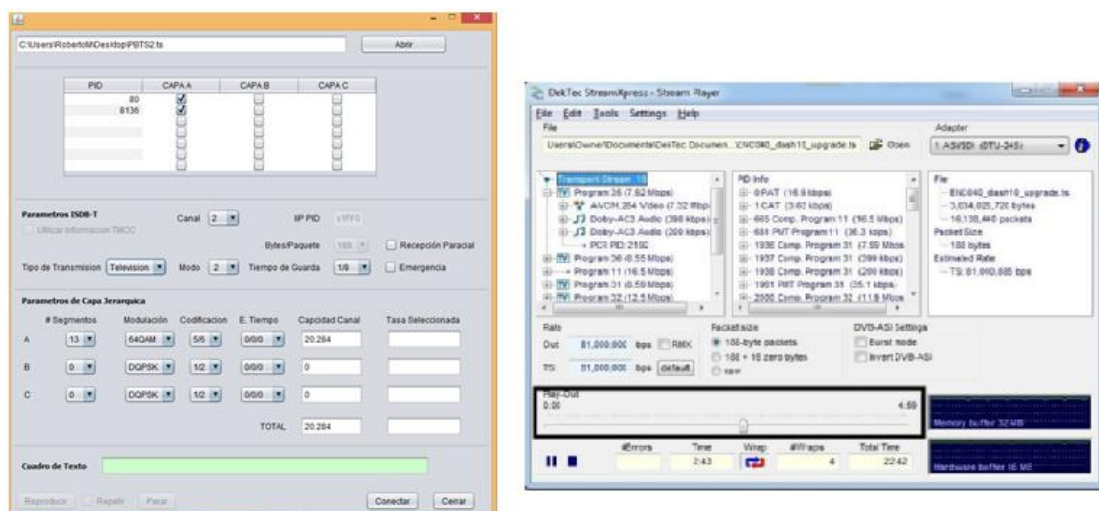


Figura 4.13 Comparación GUI Presente Proyecto y Stream Xpress

4.3.2 Transmisión de Archivos Transport Stream

Como se verificó en las pruebas ya mencionadas, el control de la tarjeta moduladora y transmisión de los archivos TS tomando en cuenta la normativa del estándar SDB-T se cumple y verifica que es igual a lo que realiza la aplicación Stream Xpress. Validamos que se puede manipular la frecuencia de transmisión, configuraciones en las capas jerárquicas y demás descritas en el bloque de transmisión tal como lo hace la aplicación modelo. La figura 4.14 muestra la frecuencia y ancho de banda utilizado en las transmisiones generadas con el software Stream Xpress, verificando que se maneja la misma frecuencia central que en el proyecto desarrollado.

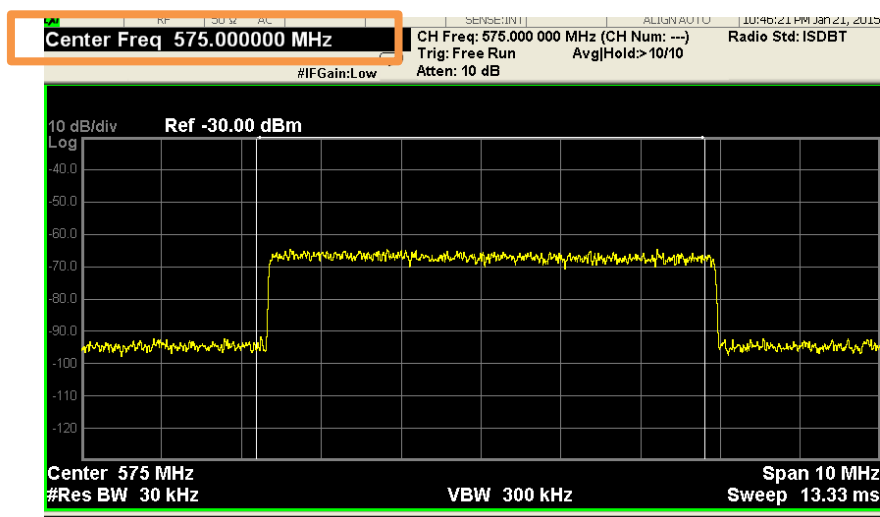


Figura 4.14 Frecuencia y Ancho de Banda de Transmisión utilizando Stream Xpress

4.4 PRUEBAS EN OTRA PC

Para permitir el uso de la aplicación en cualquier PC que contenga el sistema operativo Windows, se creó una carpeta llamada EspePlayer en la ruta /C:, dicha ruta es la de mayor jerarquía y todo PC con Windows la tiene. En dicha carpeta se ubicó el proyecto generado en Java y Visual Studio; y se modificó el directorio de inclusión adicional en el proyecto de Visual Studio, apuntando a esta nueva dirección. Se generó un archivo ejecutable utilizando la herramienta Launch4j y también se ubicó en la carpeta EspePlayer. Dichos cambios nos

permitieron utilizar la plataforma de control en distintas computadoras sin tener que realizar cambios en el desarrollo de software. Como requisitos de la PC que se va a utilizar para transmitir utilizando la plataforma desarrollada, se necesita:

- Tener instalado Windows 2007 o superior.
- Tener instalado el controlador de la tarjeta moduladora DTU-215 versión 2014.
- Tener instalado el JDK y JRE de Java para 32-bits versión 1.8.0_25 o superior.
- Cumplir con los requisitos de procesamiento y memoria que especifica DekTec para el uso de Stream Xpress mencionados en el capítulo 2..

Se realizaron pruebas en 2 computadoras adicionales donde se importó la carpeta previamente mencionada y se procedió a ejecutar la aplicación, teniendo éxito en la transmisión de los archivos TS y BTS.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Se realizó el estudio del arte investigando el estándar ISDB-T y su variable brasilera ISDB-Tb enfatizando en la capa física de un transmisor, observamos que el bloque de transmisión es igual para ambos casos.
- Se desarrolló la plataforma de control para la tarjeta moduladora de Televisión Digital DekTec DTU-215 soportando el estándar ISDB-Tb integrando las características de la tarjeta moduladora DTU-215 utilizando el SDK para Windows proporcionada por DekTec para el desarrollo del software.
- Se verificó que la librería DTAPI permite controlar todas las funcionalidades de la tarjeta como lo hace StreamXpress, permitiendo adaptar a la necesidad del usuario sin afectar el correcto funcionamiento del mismo.
- La integración de la aplicación Java y la librería C++ se logró utilizando la herramienta de desarrollo de software Java Native Interface, permitiendo en un futuro acoplar y adicionar nuevas funcionalidades al programa de acuerdo a las proyecciones de investigación del grupo de Televisión Digital

-
- Se verificó que configuraciones como frecuencia de transmisión, ancho de banda del canal, distribución de servicios en las capas jerárquicas, modulación de capas jerárquicas, etc. cumplan la normativa del estándar ISDB-Tb utilizando el analizador para televisión digital Agilent CXA
 - Se comparó el funcionamiento de la tarjeta moduladora DTU-215 siendo controlada por Stream Xpress y por la plataforma desarrollada donde se observó que ambos tienen las mismas prestaciones y no se altera el funcionamiento independiente de la aplicación.
 - Se comparó Stream Xpress y la aplicación desarrollada a nivel de interfaz gráfica de usuario, Stream Xpress presenta mayores prestaciones. Esto se da a que realiza un análisis del archivo TS que se va a reproducir, característica que está fuera del alcance del presente proyecto.

5.2 RECOMENDACIONES

- Para realizar pruebas de transmisión o incluso transmisiones hacia los usuarios, es importante pasar por una etapa de amplificación de la señal ya que las características técnicas de la tarjeta más las pérdidas de canal no permiten tener una recepción óptima de la señal.
- En la futura inclusión de nuevas funcionalidades a la aplicación sobre Java, se recomienda realizar nuevas clases, métodos e hilos tratando de evitar modificar las actuales ya que su mala modificación podría entorpecer el correcto funcionamiento de toda la plataforma.

- La actual aplicación permite únicamente transmitir los distintos programas de un archivo TS sobre la capa jerárquica A. Esto se debe a que el análisis y definición de los distintos identificadores de programa PIDs está fuera del alcance del presente proyecto. Sin embargo se habilitó un método en la librería C++ el cual, a futuro y con la integración de un analizador de TS, permite transmitir los distintos programas sobre cualquiera de las 3 capas jerárquicas seleccionadas.
- Al momento realizar la conexión de la tarjeta moduladora a la etapa de amplificación, es importante verificar que los conectores se hayan empalmado correctamente ya que su conexión incorrecta genera pérdidas que afectarán a la señal en la etapa de recepción.
- Se debe tomar en cuenta que al usar la tarjeta moduladora DTU-215 el usuario debe ser cuidadoso en el movimiento de la misma ya que la continua utilización de la misma genera un desgaste en los cables y conectores que posiblemente pueden producir pérdidas e inserción de ruido no deseado.
- El presente documento muestra un resumen de todas las funcionalidades de la librería DTAPI. En caso de querer ampliar las funcionalidades del presente proyecto se recomienda primero leer el datasheet de la tarjeta moduladora y el manual de usuario de la librería DTAPI.

BIBLIOGRAFIA

- [1] Superintendencia de Telecomunicaciones, «Compendio Histórico de las Telecomunicaciones,» 2007. [En línea]. Available: <http://www.supertel.gob.ec/pdf/publicaciones/folleto1.pdf>. [Último acceso: 15 Agosto 2014].
- [2] M. Illescas y D. Villamarin, «Implementación de un Transmisor de Pruebas de TV Digital ISDB-Tb para la Emisión de Aplicaciones Interactivas,» ESPE, Sangolquí, Ecuador, 2011.
- [3] DIBEG, «The Launching Conutry: DIBEG,» [En línea]. Available: <http://www.dibeg.org/world/world.html>. [Último acceso: 15 Agosto 2014].
- [4] Advanced Television System Committe, «ATSC Digital Television Standard, Part 1 – Digital Television System,» Washington D.C., 2013.
- [5] Digital Video Broadcasting Project, «DVB Fact Sheet,» 2014.
- [6] D. Villacrés, «Implementacion de Sistema Piloto de Transmisión de Alerta de Emergencia sobre la Television Digital Terrestre en el Ecuador,» ESPE, Sangolquí, Ecuador, 2013.
- [7] N. Granja, «Análisis del Transport Stream para el Estándar de Televisión Digital ISDB-Tb,» ESPE, Sangolquí, Ecuador, 2011.
- [8] ISO/IEC, «ISO/IEC 13818-1: Information technology — Generic coding of moving pictures and associated audio information: Systems».
- [9] ABNT, «ABNT 15601: Televisión digital terrestre - Sistema de transmisión ISDB-Tb,» 2007.

-
- [10] Kathrein, «Kathrein NDS-3542 - Encoder Modulator ISDB-T,» [En línea]. Available: <http://www.kathrein.com.br/imagen/download/nds-3542-encoder-modulador-isbd-t-21-1-2014-17-29-30-543.pdf>. [Último acceso: 19 Agosto 2014].
- [11] PROMAX, «MODULADOR ISDB-T/TB MO-370LE,» Junio 2012. [En línea]. Available: <http://www.promax.es/downloads/products/esp/MO-370le.pdf>. [Último acceso: 19 Agosto 2014].
- [12] TECSYS, «Modulador Digital ISDB-T/ TS 95000 ISDB,» [En línea]. Available: <http://www.tecsysbrasil.com.br/downloads/folders/TS9500-ISDB.pdf>. [Último acceso: 19 Agosto 2014].
- [13] Dektec Digital Video, «Catálogo 2014,» 2014. [En línea]. Available: <http://www.dektec.com/Products/Catalogue.pdf>. [Último acceso: 21 Agosto 2014].
- [14] DekTec Digital Video, «DTC-300-SP User Manual,» 2012.
- [15] DekTec Digital Video, «DTAPI – C++ API for DekTec Devices,» 2011.
- [16] D. Chapell, «INTRODUCING VISUAL STUDIO 2010,» 2010.
- [17] Oracle Java Technology, «Java SE Documentation JNI,» 25 noviembre 2014. [En línea]. Available: <http://docs.oracle.com/javase/6/docs/technotes/guides/jni/spec/intro.html#wp725>.
- [18] ISO/IEC, «ISO/IEC 13818-1: Information technology — Generic coding of moving pictures and associated audio information: Systems,» 2007.