



FACULTAD DE INGENIERÍAS Y CIENCIAS AGROPECUARIAS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL
PROCESAMIENTO DE IMÁGENES SATELITALES NOAA EMPLEANDO
MATLAB

AUTOR □

LIZETTE ESTEFANÍA ENRÍQUEZ GUERRA

AÑO

2017



FACULTAD DE INGENIERÍAS Y CIENCIAS AGROPECUARIAS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL
PROCESAMIENTO DE IMÁGENES SATELITALES NOAA EMPLEANDO
MATLAB.

Trabajo de titulación presentado en conformidad con los requisitos establecidos
para optar por el título de Ingenier□ en Redes y Telecomunicaciones

Profesor Guía

MSc. Jorge Wilson Granda Cantuña

Autor□

Lizette Estefanía Enríquez Guerra

Año

2017

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante orientando sus conocimientos y competencias para un eficiente desarrollo del tema elegido y cumpliendo con todas las disposiciones vigentes que regulan los Trabajos de titulación”

Jorge Wilson Granda Cantuña
Máster en Ingeniería Eléctrica
C.I.: 1708594187

DECLARACIÓN DEL PROFESOR CORRECTOR

Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.

Ing. Diego Fabián Paredes Páliz
Master en Comunicaciones Ópticas y Tecnologías Fotónicas
C.I.: 0603014143

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

Lizette Estefanía Enríquez Guerra

C.I.: 1721820015

AGRADECIMIENTO

Agradecer especialmente a Rosa Guerra y Marco Núñez por toda su confianza y apoyo incondicional en estos años.

Al Ingeniero Jorge Granda por la colaboración y guía durante la realización del proyecto.

DEDICATORIA

El presente trabajo lo dedico a mi hija Camila quien me motiva a seguir adelante cumpliendo todas las metas y es la razón principal en mi vida.

A mi familia y esposo quienes con su apoyo y consejos siempre me han ayudado en el transcurso de mi carrera, gracias a su apoyo incondicional han motivado cada día en mí esfuerzo y superación tanto personal como profesional.

RESUMEN

El presente proyecto consiste en el diseño e implementación de un sistema para el procesamiento de imágenes satelitales NOAA empleando MATLAB. Para el desarrollo del sistema primero se debe establecer la conexión hacia un Satélite NOAA para obtener la señal en frecuencia modulada, la señal obtenida será procesada a través de una interfaz de bloques en el software Matlab.

El proyecto utiliza dispositivos tales como una antena, un receptor de radio y un computador en el cual está instalado el programa Matlab.

Para el desarrollo del proyecto se estructuró la parte escrita de cuatro capítulos distribuidos de la siguiente manera:

Capítulo 1: Se describen los temas de investigación para el desarrollo del proyecto; una breve descripción de los sistemas Satelitales, características de la antena como medio de recepción.

Capítulo 2: En este capítulo se explica el diseño y la implementación del sistema para el procesamiento de imágenes y una estructura generalizada para dicho procesamiento en la plataforma Matlab.

Capítulo 3: Este capítulo describe la validación del mismo sobre diferentes escenarios de simulación y las pruebas de funcionamiento del sistema.

Capítulo 4: Finalmente en este capítulo se redactan las conclusiones y recomendaciones obtenidas durante el desarrollo del sistema de procesamiento.

En la sección de anexos, se adjuntan los datasheet de los dispositivos utilizados en el proyecto y código de programación.

ABSTRACT

This project involves the design and implementation of a system for NOAA satellite image processing using MATLAB. For system development the first step is to connect to a NOAA satellite signal frequency modulation, the obtained signal is processed through an interface block in the Matlab software.

The project uses devices such as an antenna, a radio receiver and a computer on which the program Matlab is installed. For this project the written part of four chapters distributed as follows is structured:

Chapter 1: Research topics for project development are described; a brief description of satellite systems, antenna and receiving means.

Chapter 2: This chapter describes the design and implementation of the system for image processing and a general framework for such processing in Matlab platform is explained.

Chapter 3: This chapter describes the validation of the prototype in different scenarios of simulation and system testing operation.

Chapter 4: Finally in this chapter the conclusions and recommendations obtained during the development of the processing system are written.

In the section of annexes, datasheet of the devices used in the project is described, as well as source code.

ÍNDICE

INTRODUCCIÓN	7
Alcance.....	8
Justificación	8
Objetivos.....	9
1. CAPÍTULO I. Marco Teórico 	
1.1 Historia de los Satélites	10
1.1.1 Satélites Artificiales.....	10
1.1.2 Tipos de Orbitas.....	12
1.1.3 Tipos de Satélites	12
1.2 Satélite Meteorológicos.....	13
1.2.1 Satélites Geoestacionarios	13
1.2.2 Satélites de Órbita Polar	14
1.3 Estación Terrena	19
1.4 Radio Definido por Software	22
1.6 Digitalización	22
1.7 Procesamiento de Imágenes.....	24
1.7.1 Procesamiento digital de imágenes.....	24
1.7.2 Imágenes en 2D	26
2. Capítulo II. Diseño e Implementación	39
2.1. Diseño del sistema para procesamiento de imágenes.....	39
2.2. Estructura del sistema.....	39
2.2.1. SDR NooElec.....	40
2.2.2. Software de Gestión	44
2.2.2.1 Orbitron.....	44
2.2.2.3 Wxtoimg	45

2.3.2 Instalación de Matlab	49
2.3.3 Entorno de Matlab.....	53
2.3.4 Procesador de Imágenes de Matlab	54
2.3.5 Diseño de la interfaz grafica	55
2.3.6 Programación	56
3. CAPITULO III. Presentación de resultados	63
3.1 Escenario de pruebas	63
3.2 Inicio de las Pruebas	71
3.3 Resultados finales	71
4. CAPITULO IV. Conclusiones y Recomendaciones ..	74
4.1 Conclusiones	74
4.2 Recomendaciones	75
REFERENCIAS.....	76
ANEXOS	79

INTRODUCCIÓN

Desde el lanzamiento del primer satélite llamado Sputnik que fue colocado en órbita en 1957 hasta la actualidad la tecnología satelital ha evolucionado notablemente para cubrir las demandas de servicios que actualmente existen.

Las áreas donde se utiliza tecnología satelital incluyen: ámbito militar, telecomunicaciones, meteorología, sistema geo referencia, debido a que se encuentran ubicados a distancias significativas sus señales llegan a la tierra con una potencia de señal muy baja. Esto ocasiona que el ruido u otro tipo de interferencias se acoplen a la señal original afectando su calidad.

En la actualidad es necesario establecer sistemas que permitan comunicaciones eficientes, el uso de la tecnología satelital ha tenido gran acogida a nivel mundial debido a su diversidad de aplicaciones. Los sistemas satelitales procesan más y más información y ello implica la utilización de equipos y de infraestructura, implementar un sistema que permita el procesamiento de imágenes satelitales es de gran utilidad debido a que se puede tener al alcance el análisis del proceso que conlleva recuperar y reconstruir la señal que contiene la información correcta.

Los sistemas establecidos hasta el momento han mejorado notablemente la perspectiva y visión que tenemos de nuestro entorno, por lo cual las imágenes que se transmiten desde el espacio hacia tierra son el pilar fundamental para el análisis y la toma de decisiones, para alertar y prevenir cualquier cambio en nuestro entorno que pueda afectar a una determinada área geográfica en la tierra.

Alcance

El alcance de este trabajo de titulación es la implementación de un sistema que permita el procesamiento de imágenes transmitidas en tiempo real por la estación satelital NOAA, la cual contiene parámetros de condiciones climáticas para un área en particular. El sistema de procesamiento de imágenes se realizará integrando una antena receptora y un software de simulación en Matlab.

En este proyecto se emplearán los conocimientos sobre sistemas satelitales adquiridos en el transcurso de los estudios de ingeniería en redes y telecomunicaciones, se identificará el proceso de transmisión y recepción de señales desde la estación satelital hasta la estación terrena. Una vez receptada la información, ésta será procesada por medio de bloques de simulación disponibles en el programa Matlab. El diseño incluirá el procesamiento de la información por medio de filtros y otros componentes con el fin de mejorar la calidad y reconstrucción de la señal original transmitida por el satélite NOAA.

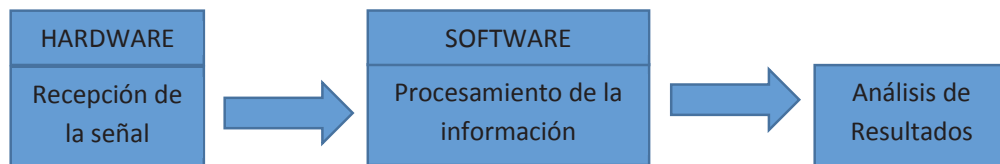


Figura 1. Arquitectura del proyecto

Justificación

El análisis del procesamiento de señales digitales es crucial para derivar información útil en diversos campos de aplicación. Mediante la implementación de este proyecto utilizando dispositivos receptores de señal y herramientas de software en cuanto a simulación se podrá realizar un análisis más cercano

a la realidad fortaleciendo el conocimiento adquirido en la carrera de ingeniería y generando un impacto a nivel investigativo en la universidad.

Es importante mencionar que para el desarrollo, éxito y aplicación del presente proyecto, el aporte de la Universidad de las Américas y de las carreras afines a Telecomunicaciones es primordial, este proyecto dará lugar a una mayor apertura a futuras mejoras de este proyecto y la creación de nuevos proyectos por parte los estudiantes en el campo de las herramientas de simulación para las telecomunicaciones.

Objetivos

Objetivo General

Implementar un sistema de procesamiento de señales satelitales NOAA utilizando Matlab y dispositivos de recepción de radiofrecuencia.

Objetivos específicos

- Identificar la operación, características técnicas de un sistema de comunicación satelital y la secuencia de pasos que se emplean en el procesamiento de imágenes desde el satélite a la estación terrena.
- Diseñar e implementar el sistema de procesamiento de imágenes satelitales utilizando Matlab y dispositivo receptor de señales.
- Generar bloques de procesamiento de señales digitales en el programa Matlab y analizar los resultados obtenidos a través del procesamiento de imágenes satelitales.

1. CAPÍTULO I. Marco Teórico

1.1 Historia de los Satélites

1.1.1 Satélites Artificiales



Figura 2. Satélite Artificial.

Tomado de: (Notario, 2013)

Según Clarke, “Planteaba la posibilidad de retransmitir información por dichos satélites hacia otros sitios de la tierra, a donde por otros medios sería imposible llegar, en forma permanente, confiable y clara.” (Clarke, 2008)

Más de cinco décadas desde el primer lanzamiento de un satélite han permitido obtener grandes beneficios de esta tecnología tales como seguridad, educación, entretenimiento y comunicaciones.

Los satélites artificiales son elementos en el espacio que orbitan alrededor del planeta Tierra mediante un vehículo espacial. El satélite está compuesto por varios subsistemas que trabajan en conjunto para su funcionamiento en el espacio.

Se menciona en la tabla 1 los siguientes subsistemas que conforman un sistema satelital

Tabla 1. Subsistemas de un Satélite

Nombre	Función
Estructura	Son los elementos y partes metálicas diseñadas apropiadamente para el funcionamiento en el espacio.
Sistema de Propulsión	Permite llegar al satélite a su órbita de destino después de ser liberado por el vehículo de lanzamiento
Sistema o subsistema de control de orientación o actitud	Su función es mantener la precisión de apuntamiento de las antenas.
Subsistema de energía eléctrica	Está formado por células fotovoltaicas que absorben los rayos solares que proporcionan de energía al sistema
Sistema de Telemetría	Está a cargo de conocer el estado de los demás subsistemas
Sistema de Telemando	Envía órdenes al satélite desde un centro de control en tierra a través de un canal dedicado.
Sistema de Control Térmico	Este sistema mantiene el margen de temperatura de funcionamiento

Adaptado de (Rosado, 2001)

Los sistemas que conforman el segmento espacial trabajan conjuntamente para el correcto funcionamiento del satélite, cada sistema y subsistema cumplen funciones específicas que permiten mantener la vida útil del satélite.

Es importante mencionar que en el espacio se obtiene la energía del sol a través de los paneles solares que proporcionan de energía a células solares, las mismas que están distribuidas en gran cantidad por el satélite.

1.1.2 Tipos de Orbitas

Los satélites están asignados a una órbita espacial definida que puede ser de tipo circular o elíptica, estas orbitas giran en el mismo sentido de la Tierra.

En la órbita geostacionaria se encuentran los satélites de comunicaciones los mismos que aparecen como un punto fijo en el firmamento sobre la superficie terrestre. Las órbitas que siguen los satélites dependen de la altura en que se encuentran.

1.1.3 Tipos de Satélites

Se puede clasificar a los satélites de acuerdo a varios parámetros, en este capítulo se describe los satélites por la órbita en la que se encuentran:

Tabla 2. Tipos de Satélite por su orbita

Tipo de Órbita	Altura sobre el nivel del Mar	Función	Satélite
LEO: Low Earth Orbit (Órbita terrestre baja)	Hasta 2,000 Km	Comunicaciones y observación de la Tierra.	Iridium Landsat
MEO: Medium Earth Orbit (Órbita terrestre Media)	Entre 2,000 y 35,000 Km	Clima Navegación GPS	Meteosat
GEO (Órbita terrestre geostacionaria)	35,789 Km Sobre el Ecuador	Satélites geostacionarios Comunicaciones Clima Navegación GPS	GOES ComSat

Adaptado de (Wikispaces, s.f.)

En la siguiente tabla se describe algunos parámetros como las bandas y rango de frecuencia en la que operan los satelitales.

Tabla 3. Bandas de Frecuencias para las comunicaciones por Satélite

BANDAS	Rango de Frecuencias GHz	de Servicio	Principales Usos
VHF	30-300 [MHz]	Fijo	Telemetría
UHF	300-1.000 [MHz]	Móvil	Navegación, Militar
L	1-2	Móvil	Emisión de radio, radiolocalización.
S	2-4	Móvil	Navegación
C	4-8	Fijo	Voz, Datos, Imágenes,TV
X	8-12	Fijo	Militar
Ku	12-18	Fijo	Voz, Datos, Imágenes,TV
K	18-27	Fijo	TV, comunicación intersatélite.
Ka	27-40	Fijo	TV, comunicación intersatélite.

Adaptado de (Gaetano, s.f.)

1.2 Satélite Meteorológicos

1.2.1 Satélites Geoestacionarios

Como su nombre lo indica se encuentran ubicados en la órbita geoestacionaria, estos satélites recorren la órbita a una altura de 36000 km y dan una vuelta en 24 horas por lo que siempre se encuentran en un punto sobre el Ecuador.

Se describe en la siguiente tabla los canales de operación del Satélite Geoestacionario

Tabla 4. Canales de Satélites GOES

Canal 1	0,55 - 0,75 μm	1 Km
Canal 2	3,8 - 4,0 μm	4 Km
Canal 3	6,5 - 7,0 μm	8 Km
Canal 4	10,2 - 11,2 μm	4 Km
Canal 5	11,5 - 12,5 μm	4 Km

Adaptado de: (Física, s.f.)

En la siguiente tabla se puede describir los parámetros de la órbita Geoestacionaria, los mismos son de gran interés para realizar cálculos de rango, ángulos de azimuth, y elevación hacia el satélite.

Tabla 5. Parámetros de una órbita Geoestacionaria Ideal

Periodo del satélite (T)	23 hr, 56 min., 4 seg
Radio de la Tierra (r)	6,377 Km
Altitud del satélite (h)	35,779 Km
Radio de la Órbita (d = r+h)	42,157 Km
Inclinación (respecto al ecuador)	0
Velocidad tangencial del satélite (v)	3.074 km/seg
Excentricidad de la órbita	0

Adaptado de: (Gaetano, s.f.)

1.2.2 Satélites de Órbita Polar

Los satélites NOAA son los más conocidos en la órbita polar, este tipo de satélites se trasladan de polo a polo y tardan aproximadamente 100 minutos en

completar una órbita alrededor de la Tierra por lo que una vuelta será en horas de la mañana y otra en la noche.

Las imágenes de los satélites (NOAA) tienen como objetivo visualizar nubes, observar el vapor de agua existente en el alta y media atmósfera, las temperaturas de la superficie de tierra y la superficial del mar.

En la actualidad están operativos los satélites NOAA-14, NOAA-15 y el METEOR 3-5, los cuales se alimentan de energía mediante paneles solares que suministran unos 200 (W) de potencia. (Senamhi, s.f.)



Figura 3 Satélite de órbita Polar

Tomado de (Senamhi, s.f.)

Tabla 6 Canales NOAA

Banda	λ	Resolución	Utilización
1	0.58-0.68 μm	1100 m	Utilización Seguimiento de nubes, nieve y hielo
2	0.72-1.1 μm	1100 m	Discriminación de las superficies de agua y vegetación
3	3.55-3.93 μm	1100 m	Temperatura superficial del mar, volcanes e incendios
4	10.3-11.3 μm	1100 m	Temperatura superficial del mar, humedad del suelo
5	11.5-12.5 μm	1100 m	Temperatura superficial del mar, humedad del suelo

Adaptado de (ocw, s,f)

En la siguiente tabla se detallan algunas características del Satélite NOAA 19.

Tabla 7. Características del satélite NOAA 19

Acrónimo		NOAA-19	
Nombre Completo	Administración Nacional Oceánica y Atmosférica - 19		
Descripción	5ª (última) unidad de vuelo de NOAA 5ª generación.		
	Misión principal: meteorología operacional.		
	Contribución significativa a la química atmosférica y al clima espacial.		
Masa en el lanzamiento	2232 kg		
Potencia	833 W		
Enlace de acceso a datos	http://www.ncdc.noaa.gov/oa/pod-guide/ncdc/docs/klm/index.htm		
Información de acceso a datos	Disponibilidad en tiempo real de AMSU-A, AMSU-B, AVHRR / 3, HIRS / 3, DCP y SEM / 2 (MEDEP + TED) por HRPT, WEFAX y DSB.		
	Tiempo de acceso diferido a GAC y LAC de archivos.		
	Mensajes de alerta de S & RSAT a LUT en tiempo real.		
Órbita	Órbita síncrona solar	Altitud	
		870 km	

Adaptado de (WMOSAT, s.f.)

Existen algunos parámetros que se deben tomar en cuenta sobre las imágenes que se obtienen de los diferentes canales del satélite. Esta información es descrita a continuación.

Utilidad de las imágenes de alta y baja resolución

Este tipo de imágenes que son de los canales APT y WEFAX tienen como finalidad la observación y se utilizan en la meteorología sinóptica, también para analizar las nubes, formas, frentes, estimaciones globales, etc.

Imágenes en el espectro visible (VIS)

La banda visible registra la radiación solar reflejada por las distintas superficies. Se representan las regiones más brillantes como nubes y hielo, mientras que las más oscuras representan el agua y el cielo sin nubes

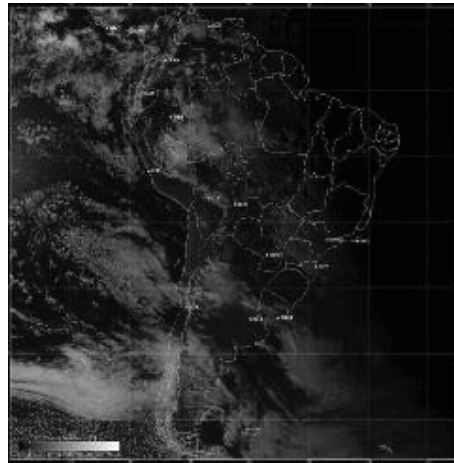


Figura 4. Canales APT y WEFAX.

Tomado (CORPAC, s.f.)

Imágenes en el espectro infrarrojo (IR).

Los satélites cuentan con equipos especiales que pueden captar la radiación térmica emitida en la tierra, estos sensores se utilizan en la noche cuando no hay radiación solar por lo cual se observan los puntos cálidos oscuros y los fríos blancos.

En este tipo de imágenes las zonas oscuras son más cálidas y las más claras son más frías.

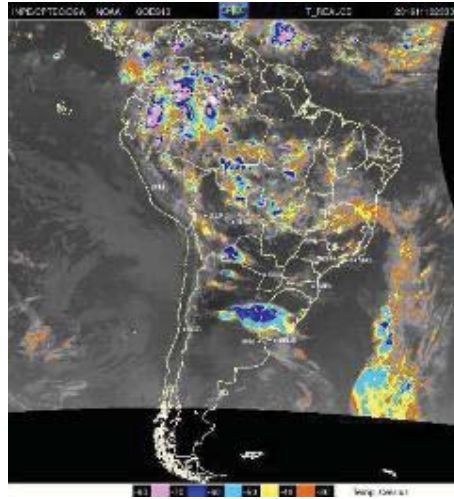


Figura 5. Imagen en infrarrojo.

Tomado  (CORPAC, s.f.)

Imágenes de vapor de agua (WV)

En estas imágenes los colores oscuros indican que el aire está seco; y los brillantes muestran que hay contenido mayor de humedad en la masa de aire.

En general el canal de WA representa la humedad media de la tropósfera

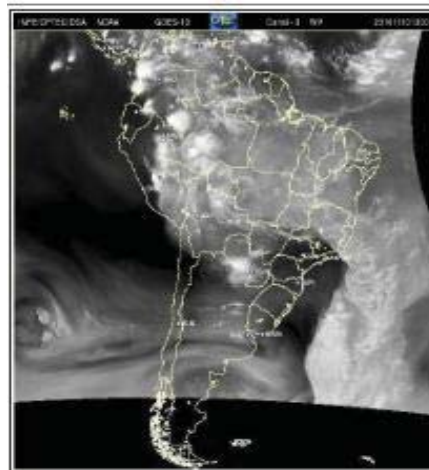


Figura 6. Imagen de vapor de agua.

Tomado  (CORPAC, s.f.)

Otros tipos de Imágenes

Las imágenes RAINSAT se obtienen mediante diversas técnicas de comparación de bandas y aplicación de matrices (obtenidas por combinación de datos de satélite y radar) y algoritmos. En estas imágenes se visualizan las zonas de intensidad de lluvia y zonas de probabilidad de precipitaciones, la temperatura del mar, del suelo o índices de vegetación, entre otros. (Senamhi, s.f.)

1.3 Estación Terrena

Se describe a una estación terrena como un conjunto de equipos interconectados entre sí con una antena o un conjunto de antenas para la recepción o transmisión de señales desde la Tierra hacia el satélite, en este proceso se debe cumplir con las funciones de modulación, demodulación, conversión de frecuencias, codificación, multiplexación, conversión analógico-digital, etc.

Como se ha mencionado anteriormente el subsistema de antenas debe cumplir con algunas características que permiten establecer la conexión hacia el satélite. Según el IEEE (*Institute of Electrical and Electronics Engineers*) define una antena como: “aquella parte de un sistema transmisor o receptor diseñada específicamente para radiar o recibir ondas electromagnéticas”. (Salleurl, s.f.)

La antena puede radiar en todas la direcciones pero esto no significa que lo haga de manera uniforme por lo que requieren de una correcta alineación (polarización). Para un sistema satelital se debe tomar en cuenta las siguientes características:

- Alta directividad, en la dirección de la posición nominal del satélite.
- Baja directividad en las demás direcciones, en especial en las cercanas al satélite.

Con las características antes descritas para la estación terrena se utiliza antenas directivas y con respecto al patrón de radiación debe tener lóbulos laterales con niveles bajos minimizando cualquier posible interferencia con otros sistemas de microondas, sean estos terrestres o satelitales.



Figura 7 Estación terrena

Tomado [] (ptolomeo, s,f)

Se puede clasificar a las antenas como direccionales y omnidireccionales; en esta clasificación se puede mencionar algunos tipos de antenas como las siguientes:

- Antenas Dipolo
- Antenas Dipolo multi-elemento
- Antenas Yagi
- Antenas Panel Plano (Flat Panel)

También es importante describir los parámetros de una antena, los cuales están descritos en la siguiente tabla.

Tabla 8. Parámetros de una antena

<i>Ancho de banda</i>	<i>Es el margen de frecuencias en el cual los parámetros de la antena cumplen unas determinadas características. Se puede definir un ancho de banda de impedancia, de polarización, de ganancia o de otros parámetros.</i>
<i>Directividad.</i>	<i>Es la relación entre la densidad de potencia radiada en una dirección, a una distancia, y la densidad de potencia que radiaría a la misma distancia una antena isotrópica</i>
<i>Ganancia</i>	<i>Es la relación entre la densidad de potencia radiada en una dirección y la densidad de potencia que radiaría una antena isotrópica, a igualdad de distancias y potencias entregadas a la antena.</i>
<i>Impedancia</i>	<i>Es la relación entre la tensión y la corriente en sus terminales de entrada. Dicha impedancia es en general compleja. La parte real se denomina resistencia de antena y la parte imaginaria, reactancia de antena.</i>
<i>Anchura de haz</i>	<i>Es un parámetro de radiación, ligado al diagrama de radiación. Se puede definir el ancho de haz a -3 dB, que es el intervalo angular en el que la densidad de potencia radiada es igual a la mitad de la máxima.</i>
<i>Polarización</i>	<i>La polarización electromagnética, en una determinada dirección, es la figura geométrica que traza el extremo del vector campo eléctrico a una cierta distancia de la antena, al variar el tiempo. La polarización puede ser lineal, circular y elíptica.</i>

Adaptado de (EduTecne, s.f.)

1.4 Radio Definido por Software

Se describe a un SDR como una tecnología donde las funciones de la capa física están definidas por software.

El objetivo de un SDR es sustituir todas las funciones analógicas por funciones digitales para conectar un equipo digital a la antena.

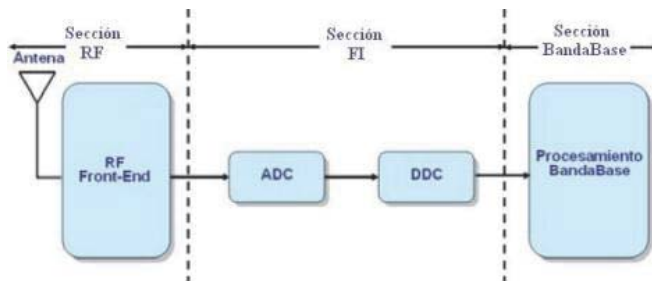


Figura 8 Diagrama de bloques receptor

Tomada de (UPM, s,f)

El rango de frecuencias para este tipo de dispositivos varía de uno a otro, pero el rango de frecuencias más comunes es de 25 [MHz] a 1,75 [GHz].

SDR es un dispositivo que cumple funciones en diferente tiempo mejorando la interoperabilidad entre diferentes servicios. Existen varias aplicaciones que se pueden realizar con el uso de este dispositivo, entre las cuales están:

- La recepción y decodificación de señales GPS,
- Permite escuchar señales de radio FM y señales satelitales
- Recepción de imágenes del satélite meteorológico NOAA.
- Escaneo de las conversaciones de radio.

1.6 Digitalización

La digitalización o conversión analógica-digital (conversión A/D) consiste en realizar de forma periódica medidas de la amplitud de una señal, y registrarlos como números enteros en cualquier tipo de memoria o soporte.

En la conversión análoga - digital se debe considerar los siguientes aspectos:

1. Muestreo: Se toma muestras periódicas de la amplitud de onda. La velocidad con que se toma esta muestra, es decir, el número de muestras por segundo, o conocido como frecuencia de muestreo.
2. Retención: Las muestras tomadas son retenidas el tiempo suficiente para permitir evaluar su nivel (cuantificación).
3. Cuantificación: Este proceso se mide el nivel de voltaje de cada una de las muestras. Consiste en asignar un margen de valor de una señal analizada a un único nivel de salida.
4. Codificación: Para este proceso se traduce los valores obtenidos durante la cuantificación al código binario, analizando que el código binario es el más utilizado.

Es importante mencionar que durante el muestreo y la retención, la señal aún es analógica, puesto que aún puede tomar cualquier valor. No obstante, a partir de la cuantificación, cuando la señal ya toma valores finitos, la señal ya es digital.

En la compresión se reduce la cantidad de datos a transmitir o grabar,

Hay dos tipos de compresión:

- Compresión sin pérdidas: Se transmite la información pero elimina la información repetida y así se ocupe menos espacio.
- Compresión con pérdidas: En este tipo de compresión al eliminar información irrelevante puede generar pérdida de calidad en el resultado final.

En la compresión gráfica JPEG se transforma la información de color a la de luminancia (1 valor por muestra) y de crominancia (2 valores por muestra) considerando que para el ojo humano es más sensible al cambio de brillo (luminancia) que de color (crominancia).

1.7 Procesamiento de Imágenes

Una imagen análoga se define como una variación de sombras y tonos, es tomada de diferentes dispositivos ópticos como telescopios o microscopios. Para que este tipo de imágenes puedan ser manipuladas en un computador, primero deben ser digitalizadas siendo la digitalización la primera acción en el procesamiento de imágenes.

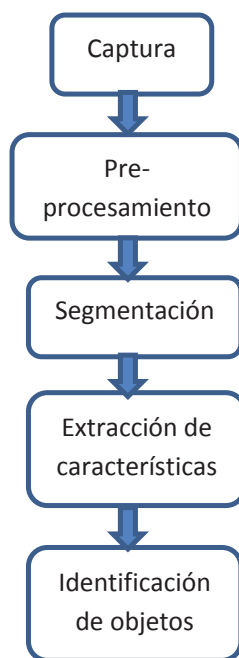


Figura 9 Etapas de procesamiento de imágenes

Adaptado de (unicen, s.f.)

1.7.1 Procesamiento digital de imágenes

Las imágenes digitales se obtienen por el proceso de muestreo y cuantificación de las señales adquiridas a través de sensores especializados sean estas cámaras u otro tipo de adquisición. El muestreo es la tarea de discretización del escenario en el plano sensor y está relacionado con la resolución espacial de la imagen.

Una imagen digital se representa en formato de grilla (matriz); M filas x N columnas donde cada elemento de la grilla es un PIXEL.

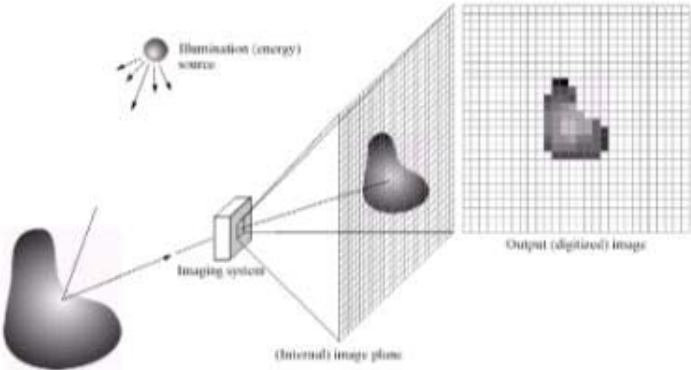


Figura 10 Captura Básica de una imagen
Tomado de (SlidePlayer, s,f)

Una vez convertida en formato digital las imágenes pueden clasificarse de la siguiente manera:

- Por dimensión: imágenes 2D y 3D
- Por paleta de colores: imágenes binarias, en grises y a color



Figura 11 Clasificación de imágenes digitales
Tomado de (SlidePlayer, s,f)

1.7.2 Imágenes en 2D

Una imagen se representa como un conjunto o matriz de píxeles de medidas $M \times N$, la imagen básica que se analiza en 2D es una imagen con escala de grises y L niveles de gris la cual está representada por la función:

$$f: [0, M-1] \times [0, N-1] \rightarrow [0, L-1] \quad (\text{Ecuación 1})$$

El proceso de cuantificación agrega una paleta de 256 niveles para el gris los cuales van desde 0 que indica negro y 255 para el blanco, con lo cual se varía la imagen en su tono total.

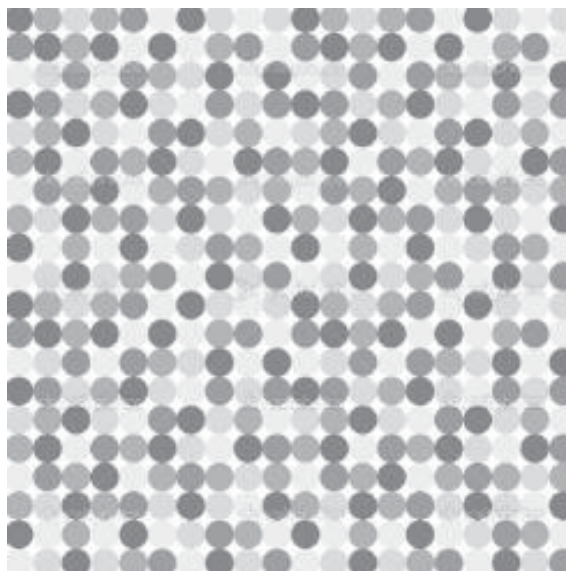


Figura 12 Escala de Grises
Tomado de (Depositphotos, s,f)

1.7.3 Imágenes a Color

En una imagen a color cada píxel está conformado por tres componentes de color diferente que son Rojo Verde y Azul o RGB, con la combinación de estos tres colores cada píxel puede tomar una tonalidad diferente de color visible en el espectro.

La luminancia (Y) es la cantidad lineal de luz, directamente proporcional a la intensidad física y ponderada por la sensibilidad al espectro de la percepción humana a la sensibilidad. Se puede calcular la luminancia como la suma ponderada de los componentes lineales: Rojo, Verde y Azul o imágenes RGB (Red Green Blue).

Los materiales de las diferentes cubiertas terrestres poseen bandas de absorción que difieren unas de otras, dependiendo de la composición química. Por ello, el procesamiento de las distintas bandas espectrales consiste en la alteración o manipulación de la información que contiene la imagen.

Tabla 9 Bandas de formato RGB

BANDA	Procesamiento
Banda 1 (azul)	Estudio de aguas, discriminación entre suelos y cuerpos de agua
Banda 2 (verde)	Vegetación, también permite realizar un estudio de la calidad del agua
Banda 3 (rojo)	Medición de absorción de clorofila, buena discriminación ente coberturas vegetales.
Banda 4 (NIR - infrarrojo cercano)	Permite delimitar la biomasa vegetal, identificación y discriminación de cultivos.
Banda 5 (MIR - infrarrojo medio)	Indicador de cantidad de agua en la vegetación Indicador de humedad de suelos
Banda 6 (infrarrojo lejano)	Incidencia hídrica, humedad de suelos, mapas térmicos.
Banda 7(MIR)	Discriminación entre tipos de rocas, geología

Tomado de (Fing, sf)

1.7.3.1 Modelo CMY

Una imagen en color no solo puede ser representada por el modelo RGB, por ejemplo para trabajos de impresión se puede usar el modelo CMY, para esto es necesario cambiar la imagen de formato RGB a CMY (cian-magenta-amarillo). La conversión se define como:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} L \\ L \\ L \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad (\text{Ecuación 2})$$

De esta conversión se define como $L+1$ la cantidad de niveles de la imagen. En figura 13 se muestra una imagen RGB en la izquierda, y a la derecha la imagen en CMY



Figura 13 Formato RGB a CMY (cian-magenta-amarillo)

Tomado de (Wikipedia, s.f)

1.7.3.2 Modelo YIQ

El modelo YIQ es el usado en los televisores y se trata de una recodificación del RGB la conversión de RGB a YIQ se define como

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{pmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (\text{Ecuación 3})$$

Y o luminancia provee la información que se necesita para señales en blanco y negro, de esta manera se puede obtener la fórmula para obtener una imagen que este en formato RGB a escala de grises.

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (\text{Ecuación 4})$$

Esta fórmula se aplica a cada pixel para lograr convertir la imagen.

1.7.3.3 Modelo HSI

Es otro de los modelos más usados es el HSI el que se caracteriza por realizar una representación más acorde a la realidad es decir trata de representar los colores tal como los percibimos los humanos, de esta manera la I representa la intensidad de la imagen, la H representa el color y la S corresponde a la saturación, las aplicaciones más usuales para este sistema son los algoritmos que usan aplicaciones para sistemas de visión humana.

Al igual que otros modelos la conversión RGB a HSI se puede calcular la componente I mediante la siguiente expresión

$$I = 1/3 * (R + G + B) \quad (\text{Ecuación 5})$$

1.7.4 Filtros

Un filtrado de cualquier tipo de señal se denomina al proceso de cambiar o modificar las amplitudes de las componentes de frecuencia, también son usados para regenerar la señal cuando hay degeneración en la misma.



Figura 14. Esquema de Filtros

Tomado de (SlidePlayer, s.f.)

Se aplican los filtros digitales para el procesamiento de imágenes utilizando métodos para resaltar o suprimir información de la imagen, durante el proceso de filtrado se aplica a cada una de las celdas de la imagen una matriz de filtrado de tamaño $N \times N$ (generalmente de 3×3) compuesta por números enteros y que actúan como coeficientes de ponderación sobre los ND de las celdas circundantes.

Las diferentes técnicas de filtrado para una imagen digital tiene como objetivos:

Tabla 10 Objetivos de filtros digitales

Suavizar la imagen	Reducir las variaciones de intensidad entre píxeles vecinos.
Eliminar ruido	Modificar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos.
Realzar la imagen	Aumentar las variaciones de intensidad, allí donde se producen
Detectar bordes	Detectar aquellos píxeles donde se produce un cambio brusco en la función intensidad.

Adaptado de (gtocom, s,f)

Se puede clasificar a los filtros digitales en dos grupos:

- **Filtros en el dominio del espacio**

En este filtro el procedimiento opera directamente sobre los píxeles y la fórmula se expresa a continuación

$$g(x, y) = T\{f(x, y)\} \quad (\text{Ecuación 6})$$

Donde

$f(x, y)$ Es la imagen de entrada

$g(x, y)$ Imagen procesada

T el operador.

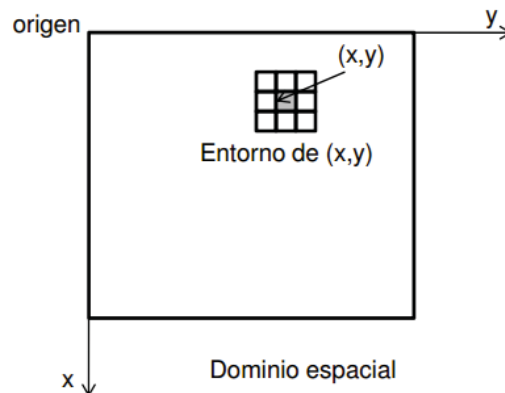


Figura 15 Vecindad de 3x3 sobre un punto (x,y)

Tomado de (Informatica, s.f.)

Dentro de los filtros de espacio de dominio se pueden dividir en dos grupos que son: Filtros lineales (filtros basados en máscaras de convolución) y filtros no lineales. Cada tipo de filtro los cuales está determinado por el contenido de la máscara.

Para la aplicación de la máscara se puede agregar ceros a la dimensión de la imagen. En la siguiente figura se puede describir la aplicación de máscara donde se toma el valor de cada píxel a partir de una operación matemática tomando en cuenta el píxel y su entorno por lo cual la convolución es la máscara que recorre la imagen píxel por píxel.

●	255	254	255	255	255	255	254	255	255
255	244	254	255	255	255	255	254	244	255
236	223	254	255	255	255	255	254	225	234
184	200	232	247	252	252	247	233	202	182
131	163	190	206	214	214	207	191	164	128
103	116	142	157	165	165	158	143	117	99
134	78	91	104	111	111	105	92	78	129
214	76	75	75	76	76	75	75	76	210
255	177	76	75	75	75	75	76	172	255
255	255	214	133	95	95	132	211	255	255

Figura 16 Aplicación de máscara

Tomado de (unicen, s.f.)

Para describir con un ejemplo tomamos una matriz de 3 x3 de la imagen en la cual se realizara cálculos con la siguiente matriz.

Tabla 11 Matriz 3x3

-1	0	1
-1	1	1
-1	0	1

Tomado de (unicen, s.f.)

Tabla 12 Matriz de calculo

75	76	210
76	172	255
211	255	255

Pixel central = $(-1)*75 + 0*76 + 1*210 + (-1)*76 + 1*172 + 1*255 + (-1)*211 + 0*255 + 1*255=530$

Tomado de (unicen, s.f.)

Dentro de los filtros lineales existe la siguiente clasificación:

- Filtro de la media

El filtro de media permite suavizar imágenes reduciendo la cantidad de variaciones de intensidad entre píxeles vecinos. Para esto asigna al pixel central la media de todos los pixeles incluidos en la ventana. La matriz de filtrado estaría compuesta por unos y el divisor sería el número total de elementos en la matriz.

Tabla 13 Máscara 3x3 para filtro de media

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Tomado de (unicen, s.f.)

- Filtros gaussianos

Se utiliza el filtro gaussiano para emborronar imágenes y eliminar ruido usando una máscara diferente la cual permite simular una distribución gaussiana.

El valor máximo aparece en el pixel central y disminuye hacia los extremos tanto más rápido cuanto menor sea el parámetro de desviación típica s . El resultado será un conjunto de valores entre 0 y 1.

Para transformar la matriz a una matriz de números enteros se divide toda la matriz por el menor de los valores obtenidos. Expresando matemáticamente con las siguientes formulas.

$$g(x, y) = e^{-x^2 + y^2 / 2 * s^2} \quad (\text{Ecuación 7})$$

$$G(x, y) = g(x, y) \cdot \min_{x,y}(g(x, y)) \quad (\text{Ecuación 8})$$

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (\text{Ecuación 9})$$

Aplicando la máscara 5x5 para el filtro gaussiano con $\sigma=1.0$ obtenemos la siguiente matriz:

Tabla 14 Matriz de 5x5

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Adaptado de (alojamientos, s.f.)

- Filtros en el dominio de la frecuencia

En el filtro de la mediana se reemplaza el valor de gris de un punto por la mediana de los niveles de gris de una cierta vecindad.

El valor final del pixel es un valor real presente en la imagen y no un promedio, de este modo se reduce el efecto borroso que tienen las imágenes que han sufrido un filtro de media y permite eliminar el ruido impulso o sal y pimienta.

La máscara $h(u, v)$ es muy utilizada en los filtros de mediana, la cual contiene los siguiente valores:

Tabla 15 Máscara $h(u,v)$

1	2	1
2	4	2
1	2	1

Adaptado de (alojamientos, s.f.)

- Filtros pasa-baja para el suavizado.

Su objetivo es suavizar la imagen para eliminar la cantidad de ruido que pueda tener la imagen. También pueden utilizarse para resaltar la información correspondiente a una determinada escala (tamaño de la matriz de filtrado)

$$G(u, v) = H(u, v) * F(u, v) \quad (\text{Ecuación 10})$$

- Filtro de media ponderada

Los elementos de la matriz de filtrado no son todos 1 sino que se da más peso a uno de ellos (generalmente el central) para obtener un resultado más parecido a la imagen original y evitar que aparezca borrosa.

Tabla 16 Filtro de media ponderada

1	1	1
1	2	1
1	1	1

Tomado de (geograf, s,f)

También podemos mencionar a los filtros para la detección de bordes, entre uno de los más utilizados es el detector de bordes de Sobel que realiza la variación entre filas y columnas.

Tabla 17 Filtros C y F de Sobel

-1.0	0	-1.0	-1.0	-2.0	-1.0
-2.0	0	2.0	0	0	0
-1.0	0	-1.0	1	2	1

Tomado de (geograf, s,f)

- Transformada de Fourier

Aplicando la transformada de Fourier solo se realiza una transformación del dominio del espacio al dominio de frecuencia sin perder información de la imagen lo cual permite aplicar algoritmos y de determinar ciertas propiedades de la imagen.

La transformada de Fourier es una función compleja y en dos dimensiones se expresa matemáticamente de la siguiente manera.

$$F(U, V) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j2\pi(Um+Vn)} \quad (\text{Ecuación 11})$$

En Matlab la función que se usa para generar los filtros en imágenes 2D se llama *fspecial* y contiene filtros preestablecidos. En la siguiente tabla se muestran los diferentes tipos de filtros en la función:

Tabla 18 Tipos de filtros

Valor	Description
'average'	Averaging filter
'disk'	Circular averaging filter (pillbox)
'gaussian'	Gaussian lowpass filter
'laplacian'	Approximates the two-dimensional Laplacian operator
'log'	Laplacian of Gaussian filter
'motion'	Approximates the linear motion of a camera
'prewitt'	Prewitt horizontal edge-emphasizing filter
'sobel'	Sobel horizontal edge-emphasizing filter

Adaptado de (mathworks, 2016)

Ejemplo de un filtro motion en Matlab

```
H = fspecial('motion',20,45);
```

```
MotionBlur = imfilter(I,H,'replicate');
```

```
imshow(MotionBlur);
```



Figura 17 Filtro motion

Tomado de (mathworks, 2016)

```
Ejemplo de un filtro disk  
H = fspecial('disk',10);  
  
blurred = imfilter(I,H,'replicate');  
  
imshow(blurred);
```



Figura 18 Filtro Disk
Tomado de (mathworks, 2016)

1.7 Software para procesamiento de Señales Satelitales

1.7.1 Matlab

Matlab es un programa que nos permite la resolución numérica de problemas a través de cálculo matricial, el proceso de señal y visualización gráfica sin utilizar lenguajes tradicionales de programación.

Es importante conocer los comandos en Matlab para el procesamiento de las imágenes; por ejemplo una imagen de color RGB se representa por tres matrices bidimensionales, correspondientes a los planos R, G y B. Se ejecutan los comandos:

```
>> Im_R = f(:,:,1)
```

```
>> Im_G = f(:,:,2)
```

```
>> Im_B = f(:,:,3)
```

Los siguientes comandos nos permiten conocer información de la imagen y se describen como se ejecutan en Matlab:

```
>> [M,N] = size (f)
```

M = número de filas, N = número de columnas

```
>> imshow (f, G)
```

Muestra la imagen f en pantalla donde G especifica el rango de intensidades. Si omitimos G, se muestra la imagen con 256 niveles de gris por defecto.

```
>> impixel(f,i,j)
```

Acceder al pixel de coordenadas (i,j) de la imagen f.

```
>> whos f
```

Información adicional de la imagen: nombre, tamaño, bytes y clase.

2. Capítulo II. Diseño e Implementación

2.1. Diseño del sistema para procesamiento de imágenes

Para iniciar con el diseño del sistema debemos identificar la secuencia de pasos para lo cual se inicia con la recepción de la señal del satélite NOAA 19, satélite que es rastreado mediante software para conocer su ubicación, a continuación, se inicia con el proceso de la señal que incluye la decodificación de la señal para obtener la imagen base de la cual se va a realizar un nuevo procesamiento mediante Matlab.



Figura 19. Diagrama de bloques

En los diferentes bloques se identifican componentes de hardware y software necesarios para el procesamiento como los siguientes

- Antena
- SDR (Software definido por radio)
- Software de interacción y procesamiento de las señales.

2.2. Estructura del sistema

Como se ha mencionado anteriormente la estación terrena es el conjunto de hardware y software que inicia con la recepción de la señal satelital hasta mostrar la señal decodificada. Para el desarrollo del sistema se realiza la

interconexión con un dispositivo SDR Software definido por radio que previamente configurado y con la interacción de otros programas permiten la correcta sincronización con el satélite simulando a la estación terrena.



Figura 20. Programas de interacción

Adaptado de (Nooelec, s,f)

La señal es captada a través de la antena y del SDR NooElec R820T el cual se conecta vía usb al computador; para obtener la señal del satélite. En la decodificación de la señal se utiliza el programa Wxtoimg el cual muestra la imagen final que está captando el satélite NOAA 19.

2.2.1. SDR NooElec

En la figura 21 se muestran tres diferentes presentaciones del dispositivo cada uno viene en un empaque conjunto con una antena de recepción básica y conexión usb 2.0 de esta manera se convierten en dispositivos de manejo muy simple y muy útil para diversas aplicaciones.



Figura 21. Dispositivos SDR

Adaptado de (Nooelec, s,f)

Dentro de la estructura del hardware del SDR se pueden identificar los siguientes bloques detallados en la figura 22.

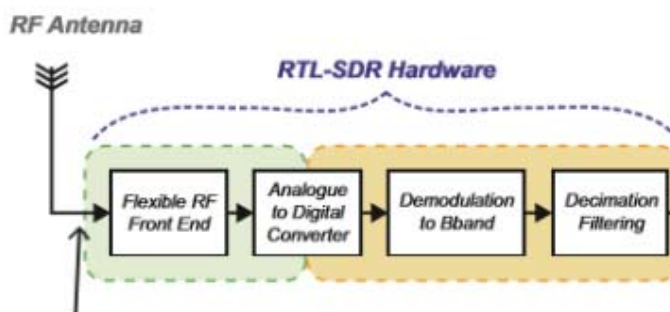


Figura 22. Estructura del hardware del SDR

Adaptado de (Mathworks, s,f)

Se muestra en la siguiente imagen el esquema del dispositivo RTL2832, este dispositivo se utilizará para realizar las pruebas durante el desarrollo del proyecto. En el esquema se visualiza los procesos de conversión de la señal de análoga a digital, el proceso digital y la interfaz de conexión USB.

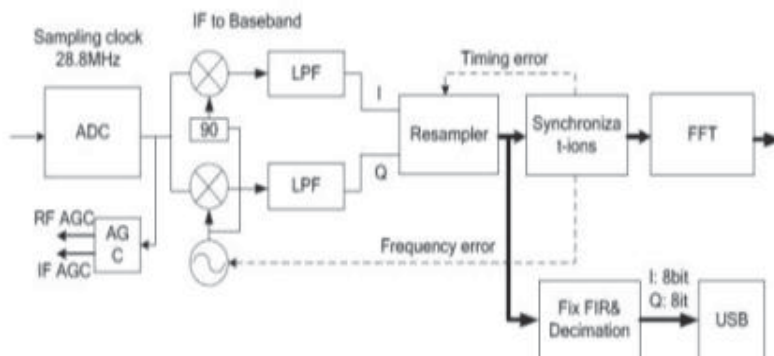


Figura 23 Esquema de RTL-SDR

Tomado de (datasheetcafe, s,f)

En el desarrollo del proyecto nos centraremos en las señales captadas alrededor de los 137 [MHz] que están asignadas para meteorología. En la figura 24 se muestra el rango de frecuencias y el tipo de señales que se pueden captar con el dispositivo.

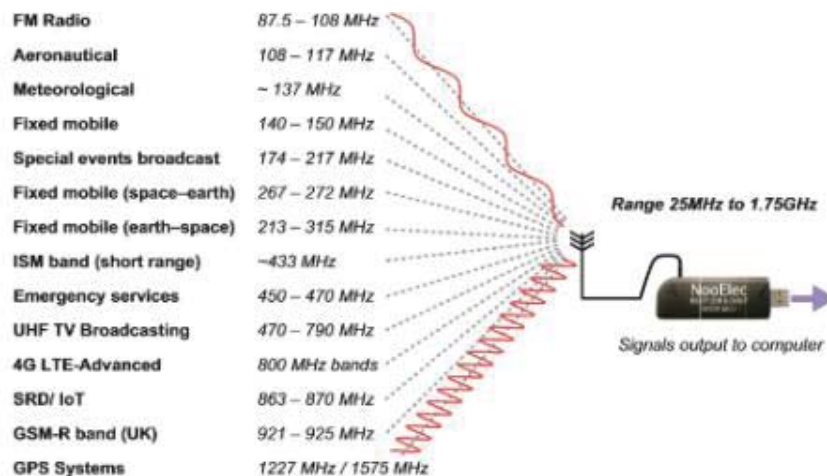


Figura 24. Rango de Frecuencias

Adaptado de (Mathworks, s,f)

El RTL-SDR incorpora un socket de conexión MCX el cual facilita la conexión para antenas omnidireccionales si necesitamos conectar otro tipo de antenas necesitaremos adaptadores de antenas de baja pérdida para poder acoplar las nuevas antenas.



Figura 25. Conector SMA

Tomado de (Seguridadwireless, s,f)

Como todo dispositivo USB que se va a conectar a un computador es necesario instalar el software para lo cual necesitamos bajar el driver que se encuentra disponible en la página principal de Nooelec una vez descargada la carpeta buscamos el ejecutable ZADIG el cual inicia la instalación del driver como se puede observar en la figura 26, se presiona el botón install driver e inicia la instalación.

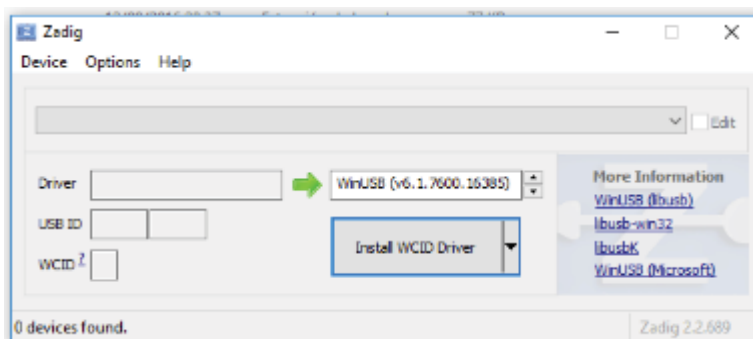


Figura 26. Programa Zadig

2.2.2. Software de Gestión

2.2.2.1 Orbitron

Es un programa que nos ayuda a verificar las orbitas de los diferentes satélites que están operando y traza su curso lo cual es beneficioso para predecir la hora en la que podemos captar la señal, es un programa muy intuitivo, ya que vamos a centrarnos en la captura de las imágenes de los satélites NOAA para lo cual tenemos que cargar la lista de los diferentes satélites NOAA misma que nos informa una serie de datos como frecuencia de Dlink, latitud, longitud, etc.

En la figura 27 para el Satélite NOAA 19 se tiene los siguientes parámetros

Longitud 156.4562" E

Latitud 18.1083"S

Altura 859.765

Y la órbita que es 39567.

NOAA 19 [+]	
Lon	156.4562° E
Lat	18.1083° S
Alt (km)	859.765
Acm	248.2°
Alt	-59.5°
AR	14h 50m 29s
Decl	-10° 39' 13"
Dist. (km)	11 971.268
RRt (km/s)	1.464
Vel (km/s)	7.421
Dirección	Ascendiendo
Eclipse	No
MA (fase)	93.4° (66)
TA	93.6°
Órbita #	39 567
Mag (ilum)	no observable
Constellation	Lib
Sol	

Figura 27 Parámetros del Satélite

En la siguiente imagen se observa en tiempo real la dirección que tomará el satélite y la huella o footprint que cubrirá en su paso.

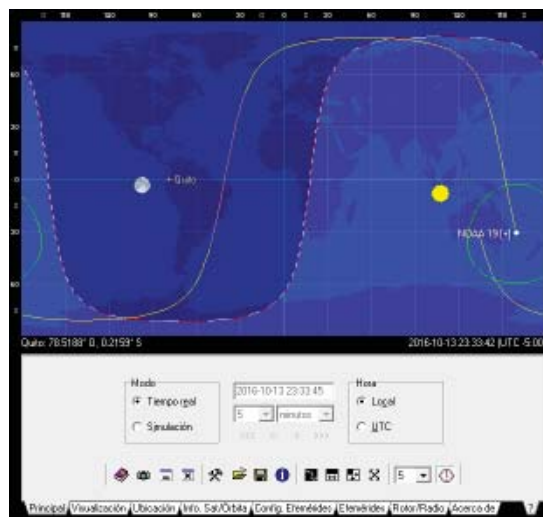


Figura 28. Interfaz de Orbitron

2.2.2.3 Wxtoimg

Para obtener el resultado de la señal captada se usa el programa WxToimg el cual decodifica la señal del satélite y muestra la imagen con la que se trabajara posteriormente. Con este programa se obtiene la imagen que el satélite está transmitiendo, dependiendo del tipo de antena y de la posición se obtendrán imágenes con mayor o menor calidad.

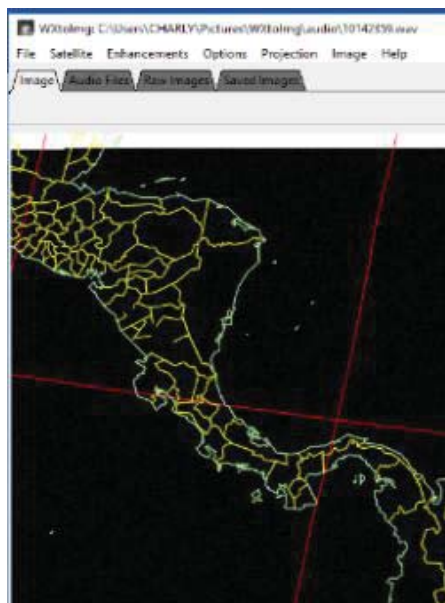


Figura 29. Señal del satélite decodificada

Para que el programa funcione se deben realizar configuraciones básicas ilustradas en la figura 30 donde se muestra la configuración de la estación terrena, aplicando la configuración con las coordenadas de la ciudad de Quito. Se ingresa la longitud y latitud del punto en donde queremos referenciar nuestra estación terrena



Figura 30. Configuración de estación local

Otra de las configuraciones es calibrar el control mix de audio, de esta manera la señal tendrá una mayor ganancia por lo que el programa puede reconocer la señal enviada por el satélite de una mejor manera.

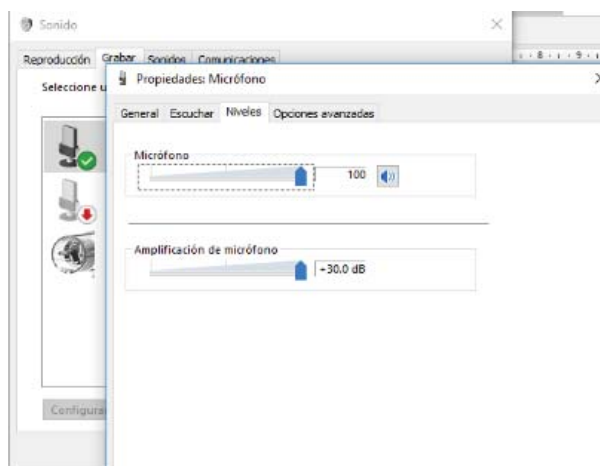


Figura 31. Configuración de volumen de dispositivo

Para lograr captar las imágenes debemos conocer las orbitas y los tiempos en los que los satélites realizan una cobertura de la estación terrena, este paso lo logramos con los programas anteriormente detallados; el programa Wxtoimg tiene dos formas de captar las imágenes; de forma manual y automática. La primera ofrece un stand by hasta que el programa recibe la señal del satélite y empieza a grabar; la forma manual inicia la grabación aun cuando no esté presente la señal del satélite,

En la figura 32 se muestra el panel de grabación para escoger la forma de grabar, cabe mencionar que el programa realiza una grabación de audio, el cual es decodificado posteriormente para la muestra de la imagen.

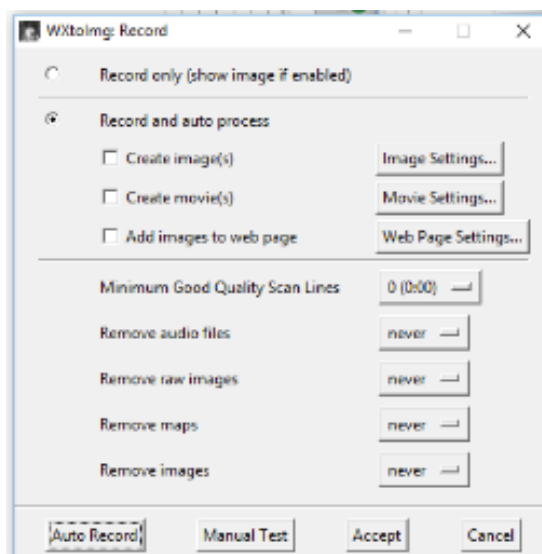


Figura 32. Consola de grabación

Los diferentes archivos guardados se almacenan en formato wav como se puede ver en la figura 33, cada uno de estos archivos puede abrirse para obtener la imagen base con la cual se va a trabajar en los siguientes pasos.

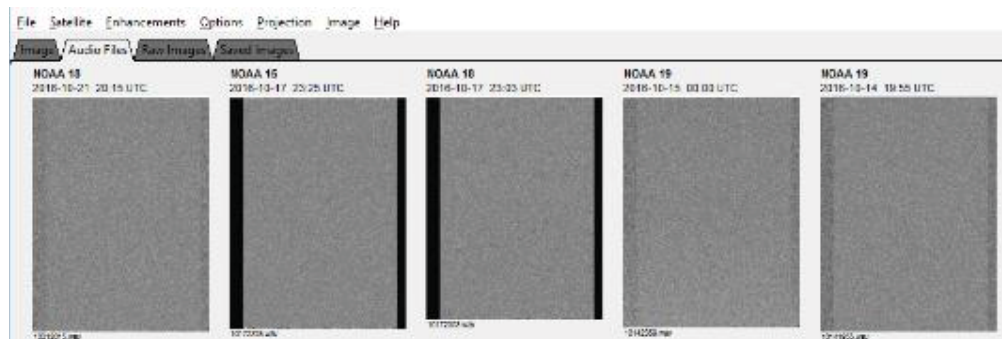


Figura 33. Captura de señales

Una vez abierta la imagen en la opción mejoras del programa se obtiene un submenú en el cual constan diferentes opciones de calibración para la imagen, estas mejoras solo están disponibles dependiendo de la calidad de la señal que se está enviando por el satélite, ya que las imágenes que se están captando son básicas y de baja resolución; no todas las opciones pueden ser aplicadas.

En la figura 34 se muestra la imagen base y el menú que se muestra para mejoras de dicha señal.

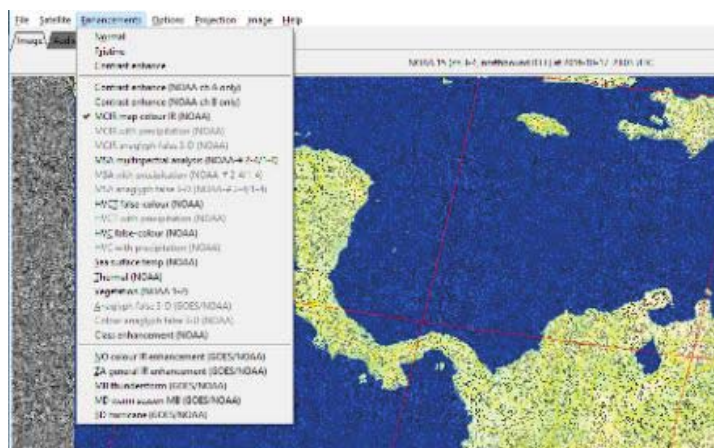


Figura 34. Parámetros de visualización

El último paso con el programa es guardar estas imágenes en nuestro computador, para ello se debe dirigir al menú file save imagen as y guardar la imagen con el nombre y directorio que se desee.

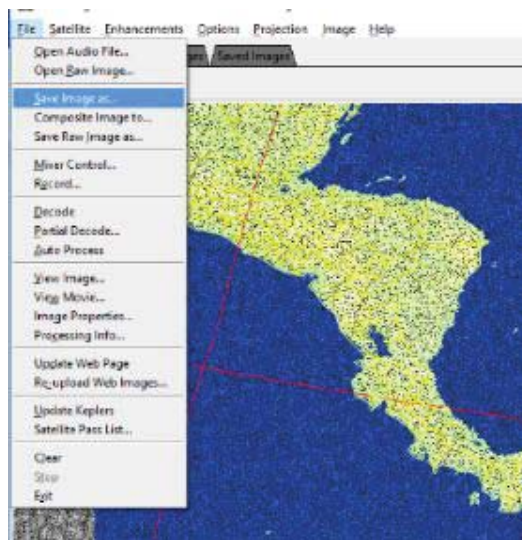


Figura 35. Opción de guardar imagen

Estas imágenes se convierten en la base para realizar el proceso de manipulación con Matlab, se las almacena en un espacio del disco duro, con esto se convierte el computador en un repositorio de imágenes.



Figura 36. Carpeta de ubicación en computador

2.3.2 Instalación de Matlab

Para a instalación del paquete Matlab podemos descargar los ejecutables en dos tipos de versiones, la primera en una versión full y la otra como un trial de 30 días. Para las dos versiones se necesita estar registrado en la página de Matlab que se encuentra en la siguiente dirección:

<https://www.mathworks.com/login>

Luego se escoge el tipo de datos para ser analizados, para finalmente descargar el instalador.

En la figura 37, se muestra la página de inicio para el login de la cuenta de Matlab

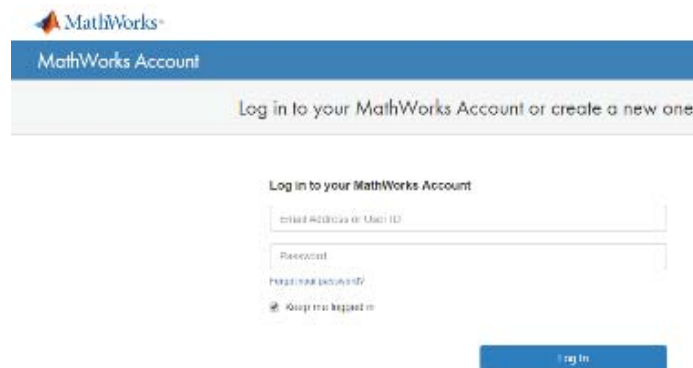


Figura 37. Log in en MathWork

Explorando la página de inicio podemos encontrar una variedad de categorías en las cuales se encuentran los productos a descargar, como se muestra en la figura 38, se tiene dos familias de productos independientes pero que se conjugan para formar un equipo de programación completo.



Figura 38. Productos y Servicios de Matlab

Finalmente, en la pestaña de Matlab se redirige al link para descargar el programa, en este punto debemos dar datos básicos para la descarga, como la plataforma en la que se está trabajando, la versión del software a descargar y el tipo de licencia que se usará.

El programa Matlab se puede instalar en varias plataformas como HP, Apple y en sistemas operativos como Windows, en este proyecto utilizaremos un computador con Windows 10. Con el instalador descargado se deben seguir los siguientes pasos para la instalación en un sistema Windows 10.

- Ejecutar el instalador que se descargó anteriormente

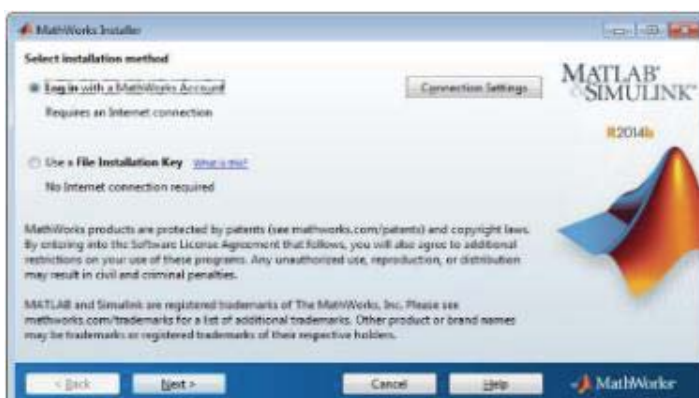


Figura 39. Método de Instalación

- Revisar la licencia y aceptar el contrato

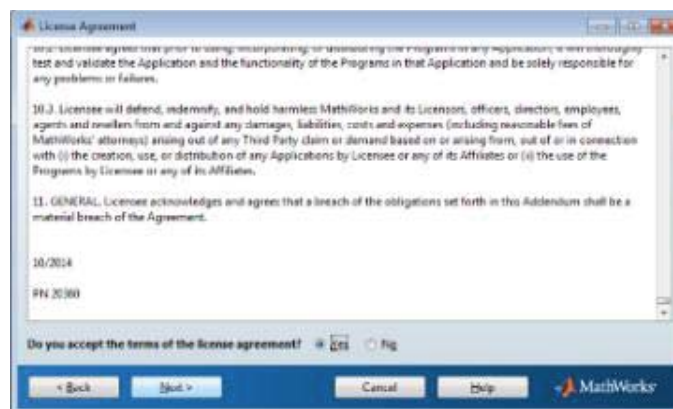


Figura 40. Aceptar Licencia

- Elegimos la carpeta para la instalación en la cual se almacenará el programa



Figura 41. Selección de carpeta

- Se elige los componentes que se van a instalar, se debe tomar en cuenta que mientras más funciones se activen, más espacio de almacenamiento va a ser necesario para la instalación.

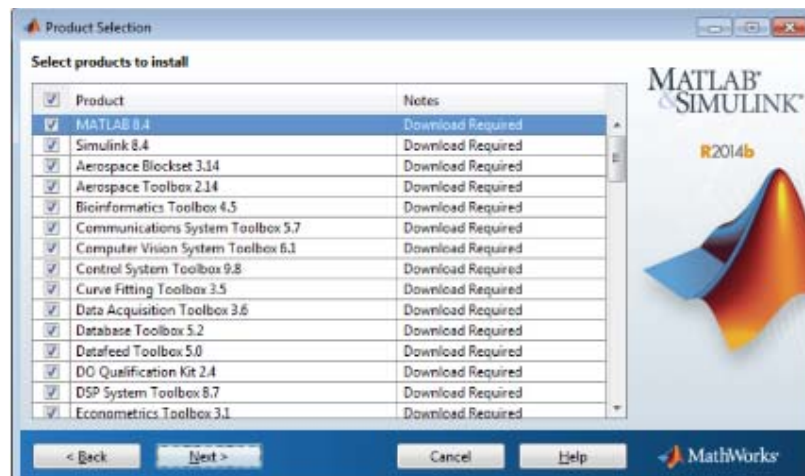


Figura 42. Selección de productos

- Finalmente se instala el programa

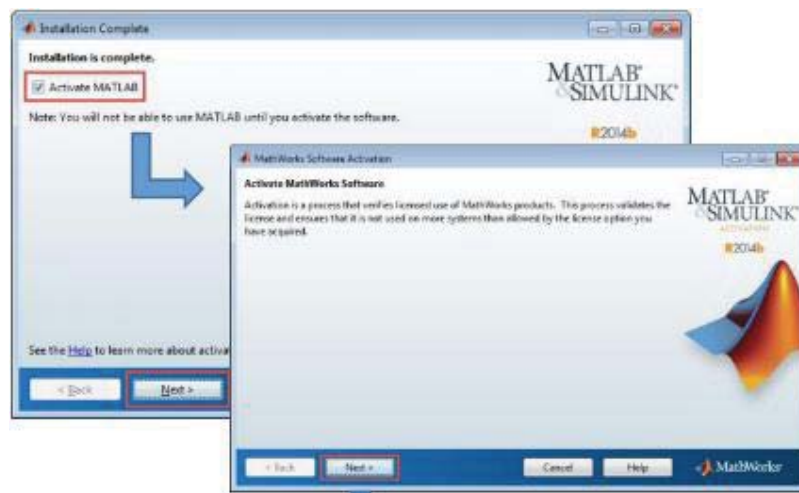


Figura 43. Activación de software

Con esto ya se tiene instalado Matlab en el computador y está listo para ser utilizado en los diferentes proyectos que se tengan para realizar.

2.3.3 Entorno de Matlab

Cuando se inicia Matlab por primera vez, se visualiza el escritorio el cual está definido por default, de esta manera es importante reconocer las diferentes áreas de trabajo las cuales pueden ser descritas como:

- Current Folder, nos muestra la carpeta actual en la que está trabajando. Es importante tomar en cuenta que los proyectos a realizarse deben ser guardados en la carpeta en la que instaló el programa, o añadir el path para poder trabajar con ese proyecto
- Command Window desde aquí se reciben las instrucciones en lenguaje de Matlab

```

Command Window
Warning: MATLAB Toolbox Path Cache is out of date and is not being used.
Type 'help toolbox_path_cache' for more info
>> SATELITE
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For more
information, click here.
>>

```

Figura 44 Ventana de Comandos

- Workspace, es donde se muestran las diferentes variables asignadas con las que se está trabajando

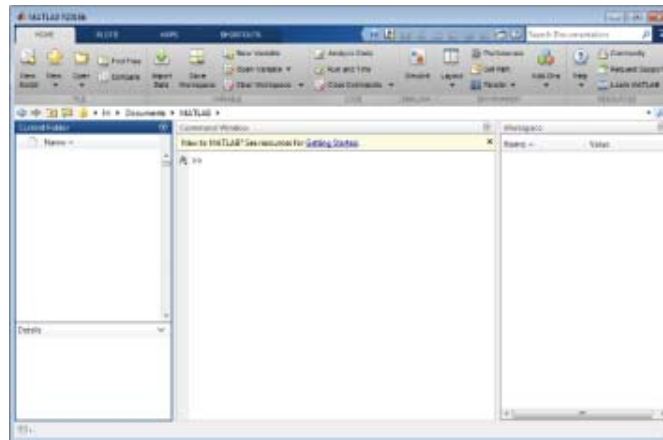


Figura 45 Espacio de trabajo para programar

2.3.4 Procesador de Imágenes de Matlab

A través de paquetes especiales, Matlab es capaz de ampliar sus funcionalidades y generar nuevos algoritmos para el proceso de análisis de imágenes, ya que una imagen digital es tratada como una matriz de puntos a los cuales se denominan pixel; Matlab es el programa más apropiado para el proceso de manipulación de imágenes.

Las funciones que se incluyen para el procesamiento de imágenes tienen las características para realizar mejoramiento de la imagen con el análisis y la aplicación de filtros.

Tratar las imágenes con diferentes propósitos es esencial en la actualidad en diferentes campos como astronomía, medicina, geofísica etc.

2.3.5 Diseño de la interfaz grafica

Puesto que Matlab es un entorno de programación se puede realizar la creación de interfaces gráficas de usuario. Para esto lo primero que se realiza es abrir el editor y con la función guide se abre el editor.

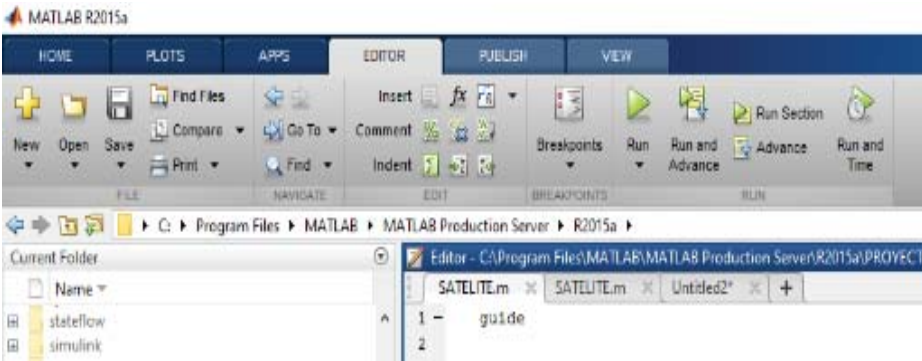


Figura 46 Interfaz gráfica de usuario GUI

Como se muestra en la figura 46 el editor consta de una tableta con los diferentes tipos de botones, menú, label etc. Y el espacio de diseño en el cual se realiza el diseño básico de la interfaz que se usará.

Cada uno de los elementos que constan en la pantalla, poseen una variedad de propiedades las cuales ayudan al momento de la programación, para acceder a las propiedades simplemente se debe hacer doble click sobre el elemento que se desee explorar.

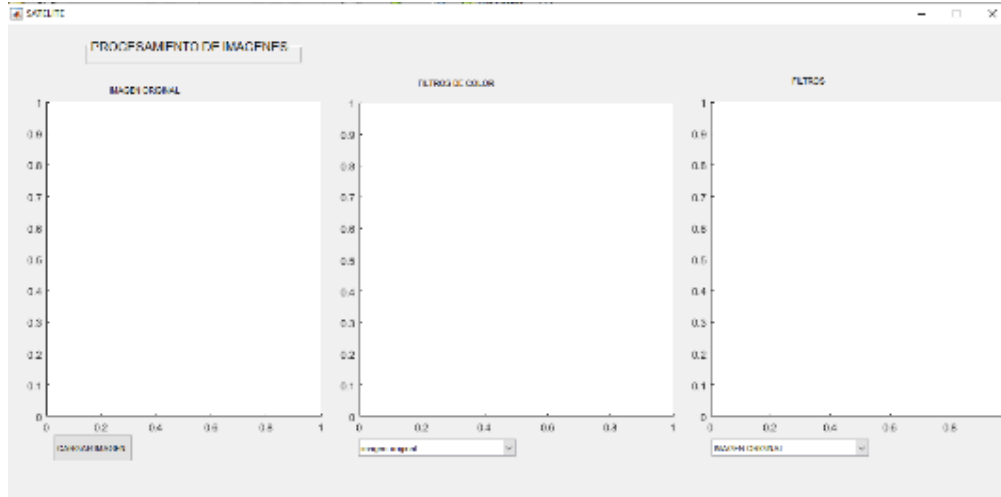


Figura 47 Interfaz con axes, label, bottom

Para el caso de los popmenu los parámetros más importantes son el Tag, nombre con el cual se identifica al elemento en la programación, el otro elemento es el String en el cual se establecen los textos que aparecen al momento que el programa está corriendo.

Entre los elementos más importantes para el diseño tenemos los Axes que son visualizadores en los que se insertan las diferentes imágenes que van a ser procesadas. Las imágenes se van a direccionar a cada uno de estos elementos en la parte de programación.

2.3.6 Programación

Una vez que se tiene el esqueleto de la GUI se debe dar sentido a los botones y al uso de filtros, en la figura 48 se muestra la GUI en la cual se realizará la programación.

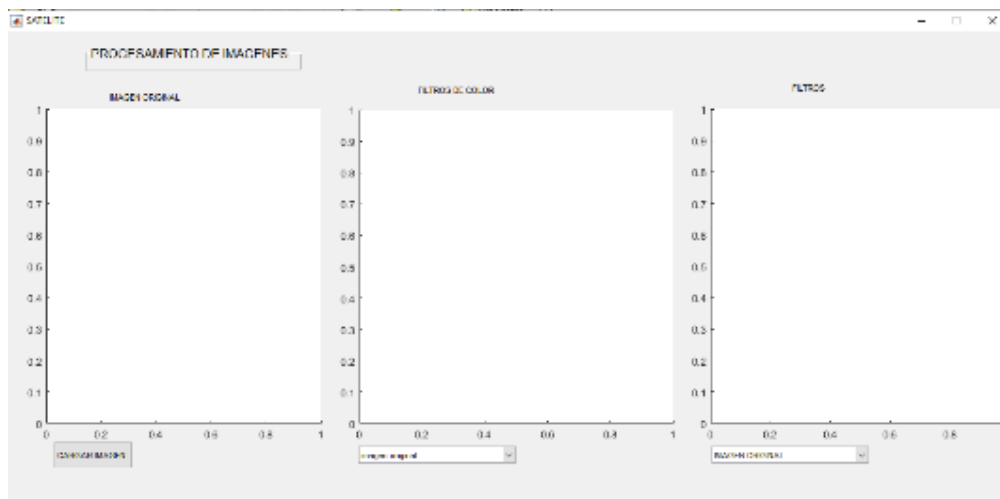


Figura 48 Esqueleto de la GUI

Se inicia con el botón para cargar las imágenes la lógica es que se pueda cargar cualquier imagen en formato JPG desde nuestra base de almacenamiento obtenidas del satélite y que puedan ser mostradas en el Axes de imagen original. Para esto se crea una función cargar imagen en la cual se coloca como parámetros abrir cualquier archivo de formato jpg y su dirección, esto se lo hace declarando una matriz con estos parámetros.

```
function cargarimagen_Callback(hObject, eventdata, handles)
[nombre dire]=uigetfile('*.jpg','Abrir');
if nombre== 0
return
end
```

El resultado es que al presionar el botón cargar imagen abrimos el directorio de archivos y se puede escoger cualquier imagen que este en formato jpg.

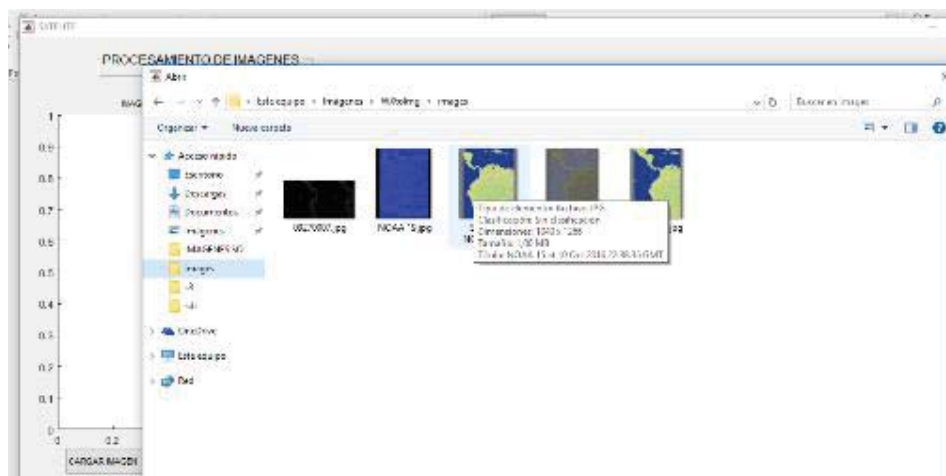


Figura 49 Directorio de archivos

Luego para que la imagen seleccionada se pueda anexar al axes de cargar la imagen se lo hace mediante el siguiente código.

```
handles.myImage=imread(fullfile(dire,nombre));
image(handles.myImage,'Parent',handles.axes1);
guidata(hObject,handles);
```

En esta parte del código lo que se está declarando es que la imagen que está siendo cargada en el paso anterior, sea cargada directamente al axes1 de esta manera tenemos la imagen base en la cual se van a realizar los diferentes procesos

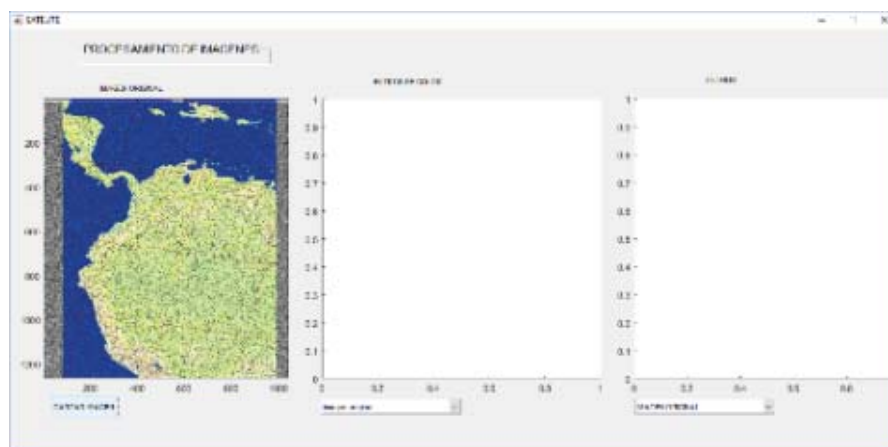


Figura 50 Cargar imagen en Axes 1

Para la programación de los dos axes restantes se usa la lógica de selección de opciones por un menú desplegable y la función que mejor puede ser aplicada es el switch la parte de programación queda de la siguiente manera:

```
guidata(hObject,handles);
v=get(handles.popupmenu1,'value');
switch v
    case 1
        imshow(handles.myImage,'Parent',handles.axes2);
    case 2
        image5=handles.myImage;
        image51(:,:,2)=image5(:,:,2);
        image51(:,:,1)=0;
        image51(:,:,3)=0;
        image(image51, 'Parent', handles.axes2);

    case 3
        image4=handles.myImage;
        image41(:,:,1)=image4(:,:,1);
        image41(:,:,2)=0;
        image41(:,:,3)=0;
        image(image41, 'Parent', handles.axes2);
```

Con esta programación lo que se está realizando es generar un bucle mediante la función Switch en la que se escogen diferentes valores del popmenu, cada uno de los casos está definido con una parte de programación que afecta a la imagen original, por ejemplo en el código anterior se definen funciones para realizar el cambio de color de la imagen en los formatos RGB, el resultado se muestra en la figura 51.

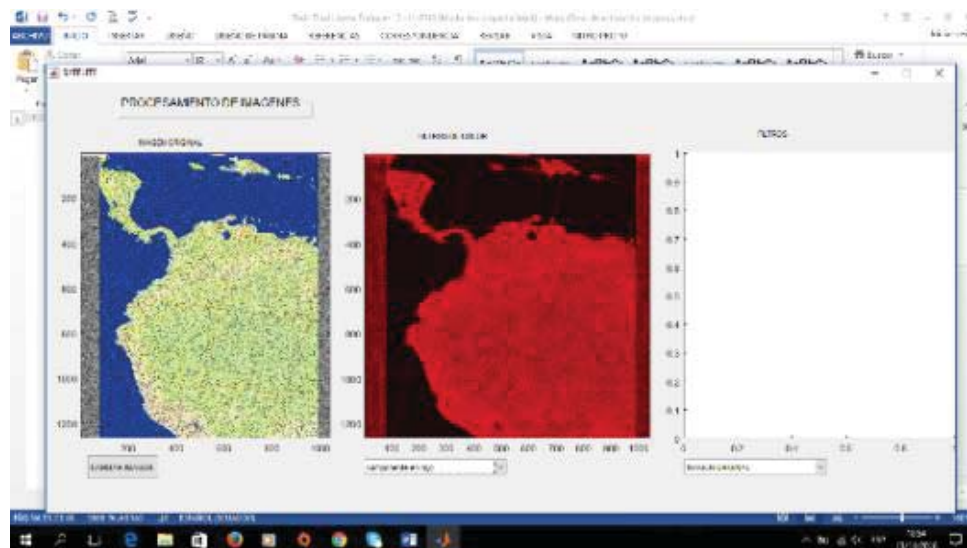


Figura 51 Aplicación de formato RGB

Para el último axes se realiza la programación con la misma lógica es decir un selector de opciones en el popmenu y que una vez procesadas a imágenes se almacenen en el axes3.

```
switch v
    case 1
        imshow(handles.mylmage,'Parent',handles.axes3);
    case 2
        im_filt=medfilt2(im_gray8);
        imshow(im_filt, 'Parent', handles.axes3);
    case 3
        filtro=1/9*[1 1 1;1 1 1;1 1 1];
        filtro_suavizado=filter2(filtro,double(im_gray8));
        image(filtro_suavizado, 'Parent', handles.axes3);
    case 4
        [a b]=imhist(im_gray8);
        stem(b,a, 'Parent', handles.axes3)
```

El resultado del código anterior se puede observar en la figura 52, en esta parte del programa se realiza el filtrado de la imagen para mejorarla.

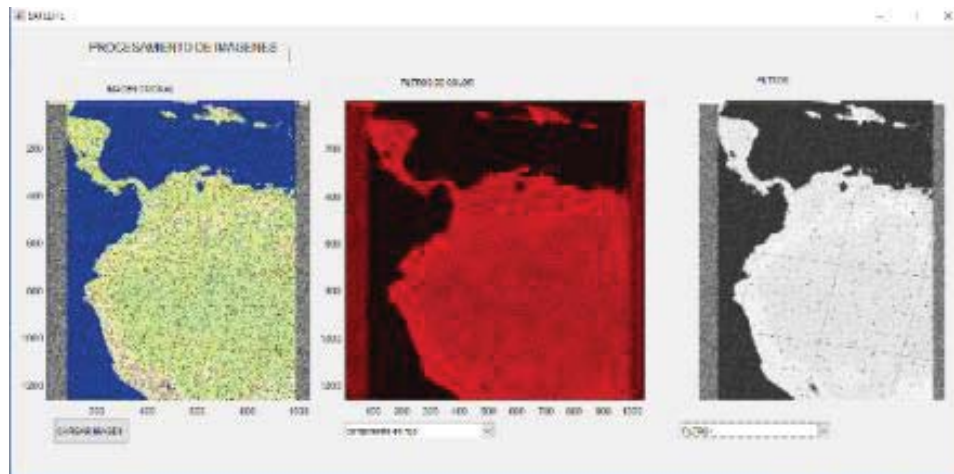


Figura 52 Filtros en Axes 3

Finalmente, como se observa en la figura 53 obtenemos el procesamiento de la imagen básica tanto en color como su mejora mediante la aplicación de filtrado. Para que las imágenes procesadas puedan ser guardadas se agregan botones para guardar cada una de las imágenes correspondientes a los axes2 y axes3 la programación es la siguiente:

```
function pushbutton3_Callback(hObject, eventdata, handles)
rgb = getimage(handles.axes2);
if isempty(rgb), return, end

formatos = {'*.jpg','JPEG (*.jpg)';*.tif','TIFF (*.tif)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR IMAGEN');
if nomb==0, return, end
fName = fullfile(ruta,nomb);
imwrite(rgb,fName);
```

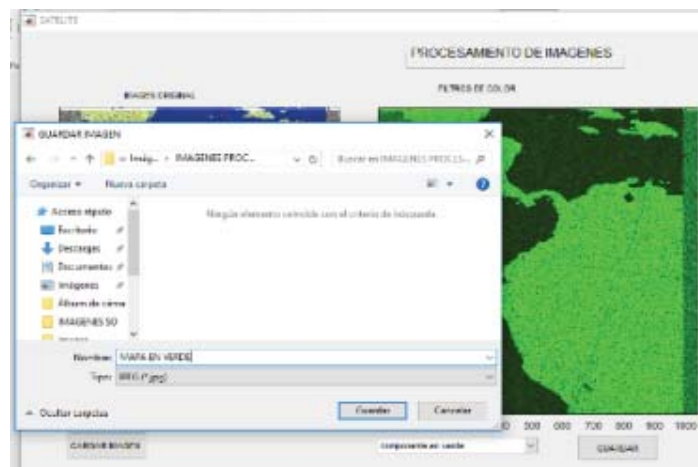


Figura 53 Botón de Guardar

De esta manera se almacena un registro de las imágenes en cualquier localidad del disco duro.

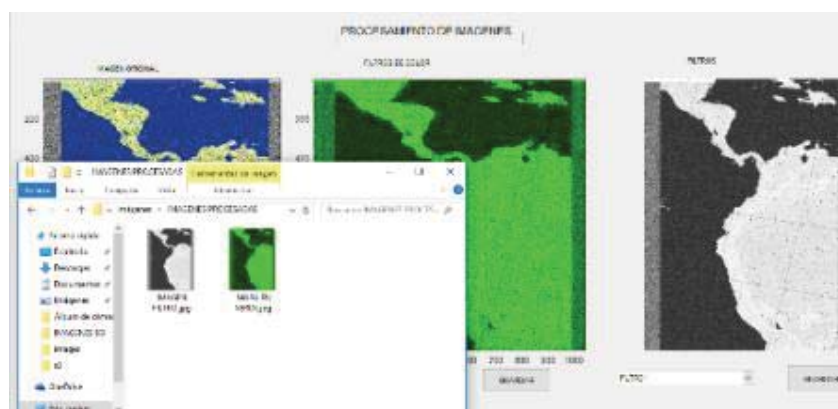


Figura 54 Registro de imagen procesada

3. CAPITULO III. Presentación de resultados

3.1 Escenario de pruebas

Las pruebas se realizaron estableciendo la conexión con el satélite NOAA que a través de su órbita cubrió nuestro punto de referencia (Quito), en diferentes horarios y fechas; con esto se puede evaluar la calidad de imágenes que se obtienen después de cada conexión.

La primera prueba consiste en verificar con el dispositivo SDR Noolec y su antena la recepción de una señal en frecuencia modulada configurando en el sintonizador los parámetros de frecuencia establecidos para el satélite. Para el satélite NOAA 19 se configura la frecuencia 137.1000 [MHz]; como segundo paso se establece una sincronización a la hora que pasará el satélite; para finalmente captar la señal y representación de la imagen.

Las pruebas consisten en dos tipos de procesos para las imágenes guardadas desde el satélite, el primero consiste en la manipulación del color y la segunda en la mejora a través de filtros. La aplicación de diferentes filtros nos permite mejorar la calidad de imagen obtenida, dependiendo de la calidad de la imagen cada filtro representará un cambio significativo. Por ejemplo podemos resaltar las nubes, vegetación, zonas cálidas y el rescate de una cordillera etc.

Tabla 19 Parámetros del sistema de procesamiento de imágenes satelitales

MENU	PROCESAMIENTO DE COLOR	PROCESO DE FILTRO
1	IMAGEN ORIGINAL	IMAGEN ORIGINAL
2	IMAGEN EN COLOR ROJO	2-D median filtering
3	IMAGEN EN COLOR VERDE	FILTRO PROMEDIO
4	IMAGEN EN COLOR AZUL	HISTOGRAMA
5	COMPONENTE BW	FILTROS COMPUESTOS
6	GRAYSCALE	N/A
7	IMAGEN CON RUIDO	N/A

Para los procesos de color se observa en la figura 56 los casos programados en el botón que controla el axes2.

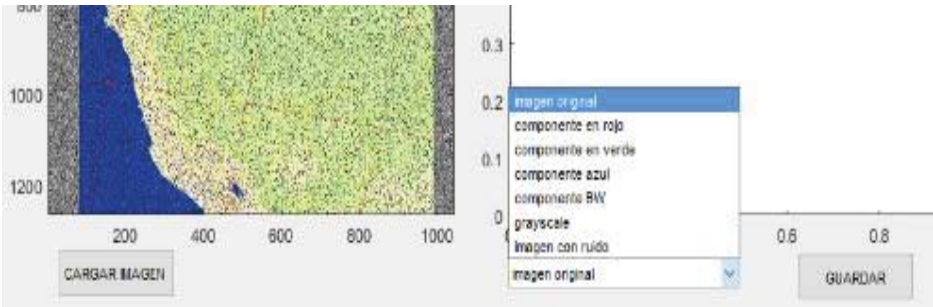


Figura 55 Opciones de procesado de imagen

Mientras que para los menús de filtros se repartieron los casos de la siguiente manera:

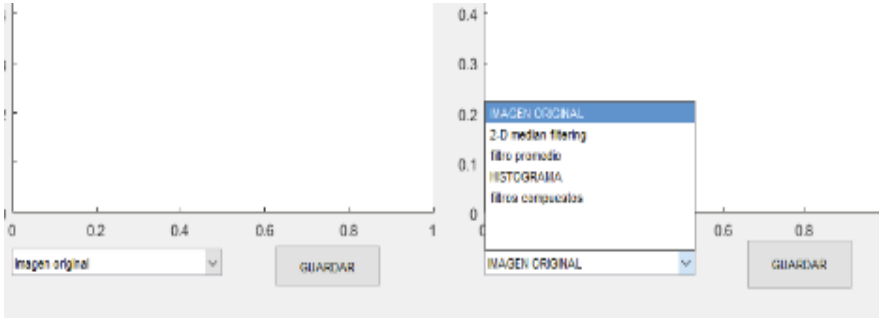


Figura 56 Menú de Filtros

Imagen original

Esta opción permite subir las imágenes que fueron capturadas del satélite y que se encuentran almacenadas en el computador. En esta opción se toma desde la apertura de la imagen fuente y se la coloca en el axes2



Figura 57 Satélite NOAA 15 tomado 10 de octubre

Imagen en color Rojo

Nos muestra la imagen con la componente en color rojo para este proceso se quitó las otras componentes de verde y azul.

Para realizar está procesamiento debemos tomar en cuenta que el color Rojo en binario se compone de la siguiente manera:

RGB (2500,0,0)= Rojo claro (o rojo brillante, o rojo intenso)

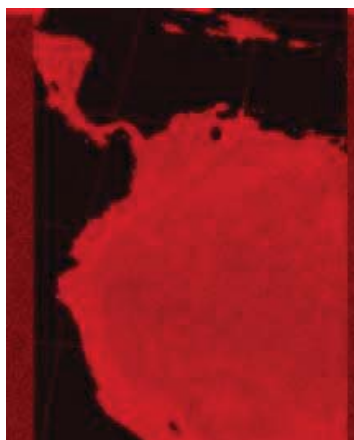


Figura 58 Componente rojo

Imagen en color verde

Al igual que el caso anterior en este caso quitamos las componentes de rojo y azul para dejar únicamente la componente de verde y mostrarla en la pantalla.

EL color verde en binario es representado como:

RBG(0,255,0) = Verde claro (o verde brillante, o verde intenso)



Figura 59 Componente verde

Imagen en color azul

Finalmente se retiran las componentes de rojo y verde para dejar solo la componente de azul.

EL color azul en binario es representado como:

RBG(0,0,255) = Azul claro (o azul brillante, o azul intenso)

Componente BW

BW = im2bw (I, nivel) convierte la imagen de escala de grises I a una imagen binaria. La imagen de salida BW reemplaza todos los píxeles en la imagen de

entrada con luminancia mayor que el nivel con el valor 1 (blanco) y reemplaza todos los otros píxeles con el valor 0 (negro).

Todos los tonos de grises del negro al blanco se representan con tres valores iguales para los componentes rojo, verde y azul (ningún color predomina), es decir que tienen la forma $RGB(x, x, x)$.

$RGB(0,0,0)$ = Negro

$RGB(128,128,128)$ = Gris oscuro

$RGB(192,192,192)$ = Gris claro

$RGB(255,255,255)$ = Blanco

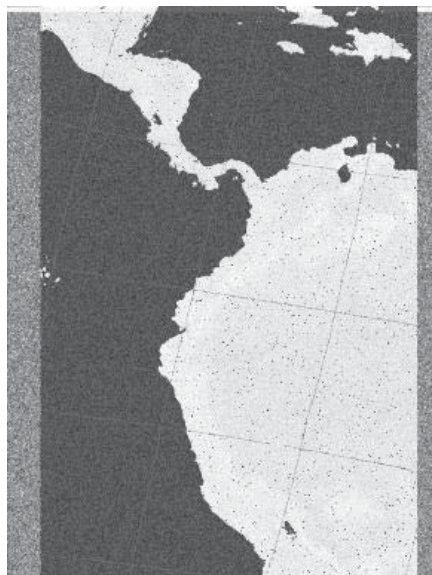


Figura 60 Componente BW

Grayscale

Convierte la imagen tricolor RGB a una imagen de intensidad de grises esta función convierte la imagen con niveles de grises conservando la luminancia de la imagen original y eliminando la información de matriz.

En Matlab se ejecuta `rgb2gray`, el cual convierte valores RGB a valores de escala de grises formando una suma ponderada de los componentes R , G y B .

$$0.2989 * R + 0.5870 * G + 0.1140 * B$$

Los valores antes descritos son los pesos mínimos para calcular el componente Y.

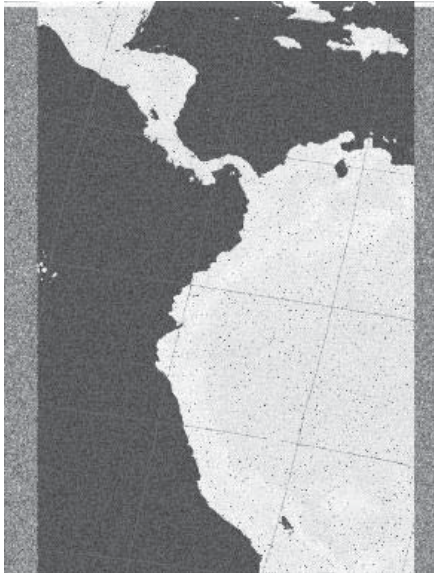


Figura 61 Filtro de Escalas de grises

Imagen con Ruido

Añade ruido a la intensidad de la señal; estas funciones se usan para simular el ruido generado en la transmisión de señales en la vida real, esta función puede tener algunas variantes como:

Tabla 20 Parámetros de ruido

<i>Valor</i>	<i>Descripción</i>
'gaussian'	Gaussian white noise with constant mean and variance
'localvar'	Zero-mean Gaussian white noise with an intensity-dependent variance
'poisson'	Poisson noise
'salt & pepper'	On and off pixels
'speckle'	Multiplicative noise

Adaptado de (mathworks, 2016)

2D Median Filtering

Como su nombre lo indica genera un filtrado medio de la matriz de pixeles de dos dimensiones cada pixel de salida tiene un valor medio de 3x3 alrededor de cada pixel.

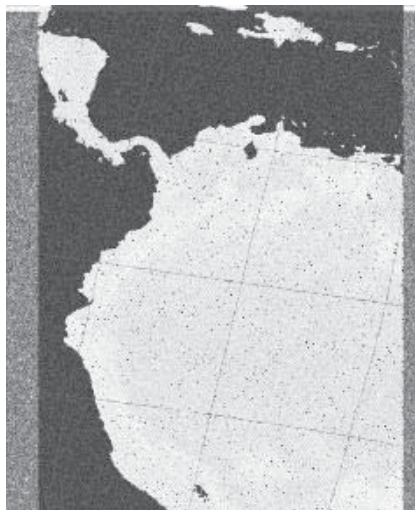


Figura 62 2D median filtering

Filtro Promedio

Obtiene el valor promedio de los pixeles, el efecto que tiene a la salida del filtro es una imagen suavizada, para usar este filtro se genera una matriz, la cual afecta a cada pixel en un promedio de $1/9$



Figura 63 Filtro Promedio

Histograma

Calcula el histograma para la imagen de intensidad I y muestra un gráfico del histograma. El número de bandejas en el histograma está determinado por el tipo de imagen.

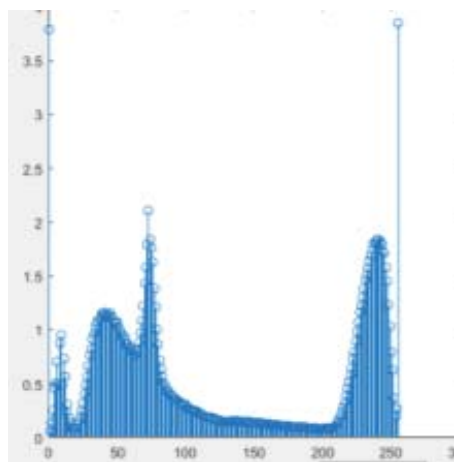


Figura 64 Histograma

Filtros Compuestos

Genera una imagen filtrada mediante un filtro medio y un filtro promedio, de esta manera se observa que una imagen puede ser filtrada n veces para obtener mejoras en la imagen.

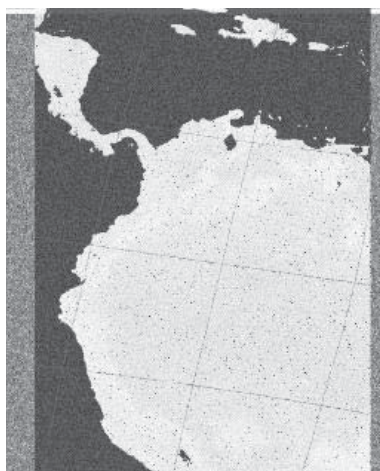


Figura 65 Filtro compuesto

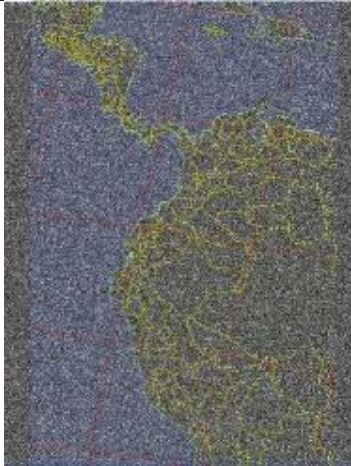
3.2 Inicio de las Pruebas


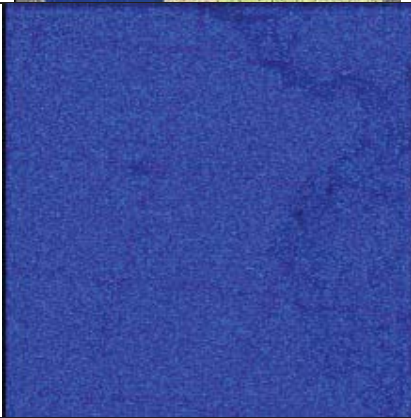
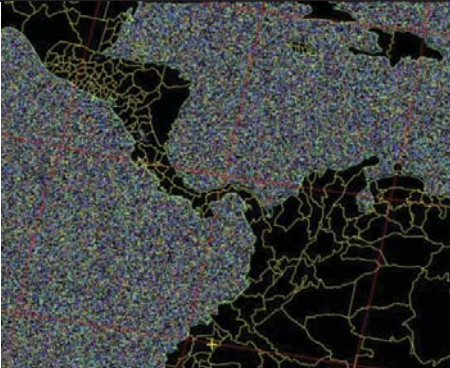
A continuación, se detallan las imágenes capturas en cada conexión con las que se realizaron en las pruebas de funcionamiento. Las pruebas para el NOAA 19 se las puede realizar solo una vez al día debido a la órbita que tiene el satélite, además se cuenta con un espacio de solo 10 minutos para captar las imágenes, una vez captadas las imágenes estas fueron almacenadas para posteriormente generar los diferentes procesos con Matlab.

3.3 Resultados finales

En esta fase se realizaron varias pruebas tanto con las antenas que originalmente vienen con el dispositivo SDR y otro tipo de antenas. Como primera parte se conectó el SDR con su antena al computador el día 27 de septiembre, una vez sincronizado con el Satélite NOAA 19 se obtiene la siguiente imagen.

Tabla 21. Resultados finales

NOMBRE	Fecha Captura	IMAGEN	OBSERVACIÓN
NOAA 19	27 de septiembre		Bastante Ruido debido a la poca ganancia de la antena.

NOAA 15	10 octubre		Se percibe menos ruido
NOAA 19	15 de octubre		Imagen totalmente distorsionada
NOAA 15	31 de Octubre		Se visualiza el perfil del continente sin embargo el ruido afecta la visualización del mar

Finalmente y después de realizar pruebas en diferentes días y con los componentes existentes obtenemos resultados variados con respecto a la calidad de la imagen que depende principalmente con la ganancia de la antena. Las señales que fueron capturadas presentan bastante ruido lo que no permite visualizar una imagen clara.

Con la aplicación de los filtros anteriormente descritos hemos proporcionado una mejora en la calidad de cada imagen de acuerdo a un requerimiento específico. Finalmente podemos concluir el mejoramiento de la calidad de la imagen resalta algunos aspectos relevantes sobre la imagen obtenida del satélite.

4. CAPITULO IV. Conclusiones y Recomendaciones

4.1 Conclusiones

Las señales captadas del satélite NOAA 19 corresponden a imágenes de baja resolución, no se pueden captar señales de mejor calidad debido a las limitaciones del rango de frecuencia del dispositivo.

El dispositivo SDR RTL2832U requiere previamente el reconocimiento con el programa Zading para que se establezca la conexión con el sintonizador FM desarrollado en Matlab.

Para captar la señal del satélite se debe configurar la frecuencia establecida en la que va a transmitir y después sincronizar con la hora local y hora UTC para grabar la señal FM.

Se puede concluir después de varias pruebas que las señales en frecuencia modulada que se obtienen del Satélite NOAA presentan ruido debido a que el dispositivo SDR trabaja en la frecuencia de 25[MHz] - 1750[MHz].

Para que el dispositivo SDR pueda establecer la conexión con el programa, en Simulink se debe descargar la librería RTL-SDR Radio la cual permite que el equipo USB se conecte al bloque de entrada.

Para la realización de estas pruebas hay que tomar en cuenta la fecha y hora en que el satélite estará cubriendo el punto como estación local que fue configurado con anterioridad, también el tiempo que estará en órbita el satélite ya que están aproximadamente 10 minutos.

Finalmente se puede concluir que con un programa de procesamiento de imágenes que integre varios filtros se puede mejorar la calidad de la imagen satelital.

4.2 Recomendaciones

Se recomienda el uso de antenas de diferente ganancia y en áreas de espacio libre, ya que el satélite se encuentra en órbita un promedio de 10 minutos con lo que hay que ser precisos al momento de captar la señal.

Se recomienda el uso de un computador con procesador Core i3 o superior ya que las simulaciones demandan gran capacidad de procesamiento lo que provoca que los equipos se pongan lentos

Usar acoples de señales que tengan bajas pérdidas ya que de por sí la señal satelital es baja y malos conectores introducen pérdidas en el sistema lo que permitió encontrar el software indicado

Se recomienda que para la etapa de programación tanto en Matlab como Simulink se definan los conceptos necesarios de programación, también conocer la lógica de programación para crear una GUI como diagrama de bloques.

Para la etapa de pruebas se recomienda utilizar una amplia variedad de antenas, las mismas que permitirán realizar varias pruebas conectándose a cada satélite y así determinar de manera más selectiva que elemento es el más idóneo para minimizar el ruido en la señal.

REFERENCIAS

- Clarke, A. (2008). *Teoría de Arthur Clarke*. Recuperado el 07 de noviembre de 2016, de <https://www.itu.int/itu-news/manager/display.asp?lang=es&year=2008&issue=03&ipage=Arthur-Clarke&ext=html>
- CORPAC. (s.f.). *Parametros de imagenes satelitales*. Recuperado el 08 de noviembre de 2016, de <http://www.corpac.gob.pe/app/Meteorologia/tiempo/satelitales.php>
- Depositphotos. (s,f). Escala de grises. Recuperado el 13 de noviembre de 2016, de <http://sp.depositphotos.com/41085699/stock-illustration-seamless-greyscale-pattern.html>.
- Edutecne. (s.f.). *Parámetros de una antena*. Recuperado el 11 de noviembre de 2016, de http://www.edutecne.utn.edu.ar/wlan_frt/antenas.pdf
- El cajon de la verdad. (s.f.). *Antena Yagi-Uda*. Recuperado el 06 de noviembre de 2016, de <http://elcajondelaverdad.blogspot.com/2013/02/antena-yagi-uda.html>
- FING. (s.f.). Percepción Remota y Procesamiento de Imágenes. Recuperado el 26 de noviembre de 2016, de <https://www.fing.edu.uy/inco/cursos/sig/clases/procimagen061011.pdf>
- Fisica. (s.f.). *Satelites Meteorologicos*. Recuperado el 24 de noviembre de 2016, de http://meteo.fisica.edu.uy/Materias/climatologia/peactico_climatologia_2010/Satelites%20Meteorologicos.pdf
- Gaetano, O. (s.f.). *Los Satélites de comunicaciones*. Recuperado el 16 de noviembre de 2016, de <http://www.casanchi.com/ast/satelitecom01.pdf>
- Huidobro, M. (2013). Antena Parabólica. Recuperado el 17 de noviembre de 2016, de http://www.acta.es/medios/articulos/ciencias_y_tecnologia/020001.pdf
- MathWorks. (s.f.). Select Your Country. Recuperado el 15 de Noviembre de 2016, de https://www.mathworks.com/help/matlab/functionlist.html?s_cid=doc_ftr
- Nooelec. (s,f). Software definido por radio. Recuperado el 11 de noviembre de 2016, de <http://www.noelec.com/store/nedr-smart-hf.html>.

- Notario, E. (2013). *Satélite Artificial*. Recuperado el 31 de octubre de 2016, de <http://es.gizmodo.com/la-revolucion-silenciosa-de-los-satelites-947867835>
- OCW. (s.f.). Canales NOAA. Recuperado el 17 de noviembre de 2016, de <http://ocw.um.es/ciencias/herramientas-de-teledeteccion-aplicadas-a-la/material-de-clase-1/tema-5-plataformas-sensores-y-canales.pdf>
- Pentagono. (s.f.). *Tutorial de Matlab*. Recuperado el 16 de Noviembre de 2016, de http://pentagono.uniandes.edu.co/tutorial/Matlab/tutorial_matlab.pdf
- PISOS. (s.f.). *Cómo orientar tu antena parabólica*. Recuperado el 10 de noviembre de 2016, de <http://www.pisos.com/hogar/bricolaje/tus-reformas/trucos-de-bricolaje/como-orientar-tu-antena-parabolica/>
- ptolomeo. (s.f.). Radio Definido por Software . (s.f.). Recuperado el 15 de noviembre de 2016, de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/894/A4.pdf?sequence=4>
- Rosado, C. (2001). *Comunicación por Satélite*. México:: Limusa, S.A. de C.V.
- Salleurl. (s.f.). *Tipos de Antenas* . Recuperado el 14 de noviembre de 2016, de http://www.salleurl.edu/semipresencial/ebooks/ebooks/ebook_teoría_antenas.pdf
- Sánchez, A. (2017). La Radio Definida por Software: (Diseño de un receptor de banda aeronáutica VHF. Recuperado el 22 de noviembre de 2016, de <https://upcommons.upc.edu/bitstream/handle/2117/83161/Resum.pdf?sequence=2&isAllowed=y>
- Senamhi. (s.f.). Tiempo y clima. Recuperado el 12 de Noviembre de 2016, de <http://www.senamhi.gob.pe>
- Seguridadwireless. (s,f). Galería de conectores. Recuperado el 21 de noviembre de 2016, de <http://www.seguridadwireless.net/hwagm/galeria-conectores.html>.
- Senamhi. (s.f.). *Satelites NOAA*. Recuperado el 21 de Noviembre de 2016, de http://www.senamhi.gob.pe/main_down.php?ub=mmt&id=cap12
- SlidePlayer. (s,f). Procesamiento digital de imágenes. Recuperado el 10 de noviembre de 2016, de <http://slideplayer.es/slide/1641157/>.
- TDXCAT. (s.f.). *Periodo del satélite NOAA*. Recuperado el 13 de Noviembre de 2016, de

<http://tdx.cat/bitstream/handle/10803/6839/05Nm05de17.pdf?sequence=6>

UCM. (s.f.). *MANUAL BÁSICO* . Recuperado el 13 de noviembre de 2016, de http://www.sisoft.ucm.es/Manuales/MATLAB_r2006b.pdf

UPM. (s.f.). Implementación y configuración de un receptor de radio definido por software (SDR) para estudios de propagación. Recuperado el 04 de noviembre de 2016, de http://oa.upm.es/22874/1/INVE_MEM_2012_153627.pdf

Wikipedia. (s.f.). Formato RGB a CMY. Recuperado el 08 de noviembre de 2016, de https://es.wikipedia.org/wiki/Modelo_de_color_CMYK.

Wikispaces. (s.f.). *Satelites Artificiales*. Recuperado el 7 de noviembre de 2016, de <https://satelites-artificiales-y-contaminacion-espacial.wikispaces.com/file/view/Actividad+2+Sat%C3%A9lites+artificiales.pdf>

WMOSAT. (s.f.). Características del satélite NOAA 19. Recuperado el 15 de Noviembre de 2016, de <https://www.wmo-sat.info/oscar/satellites/view/341>

WNDW. (s.f.). *Tipo de antenas*. Recuperado el 10 de noviembre de 2016, de <http://wndw.net/pdf/wndw-es/chapter4-es.pdf>

ANEXOS

ANEXO A

Código fuente del programa de procesamiento de imágenes

```
function varargout = SATELITE(varargin)
% SATELITE MATLAB code for SATELITE.fig
%   SATELITE, by itself, creates a new SATELITE or raises the existing
%   singleton*.
%
%   H = SATELITE returns the handle to a new SATELITE or the handle to
%   the existing singleton*.
%
%   SATELITE('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in SATELITE.M with the given input
arguments.
%
%   SATELITE('Property','Value',...) creates a new SATELITE or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before SATELITE_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to SATELITE_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help SATELITE

% Last Modified by GUIDE v2.5 10-Nov-2016 14:55:24

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @SATELITE_OpeningFcn, ...
                  'gui_OutputFcn', @SATELITE_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```

% End initialization code - DO NOT EDIT

% --- Executes just before SATELITE is made visible.
function SATELITE_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to SATELITE (see VARARGIN)

% Choose default command line output for SATELITE
handles.output = hObject;
handles.myImage=[];

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes SATELITE wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = SATELITE_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cargarimagen.
function cargarimagen_Callback(hObject, eventdata, handles)
[nombre dire]=uigetfile('*.jpg','Abrir');
if nombre== 0
    return
end
% hObject    handle to cargarimagen (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.myImage=imread(fullfile(dire,nombre));
image(handles.myImage,'Parent',handles.axes1);

guidata(hObject,handles);

```

```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox1 contents as cell
array
%         contents{get(hObject,'Value')} returns selected item from listbox1

% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listbox2.
function listbox2_Callback(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox2 contents as cell
array
%         contents{get(hObject,'Value')} returns selected item from listbox2

% --- Executes during object creation, after setting all properties.
function listbox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents
as cell array
%    contents{get(hObject,'Value')} returns selected item from popupmenu1
guidata(hObject,handles);
v=get(handles.popupmenu1,'value');
switch v
    case 1
        imshow(handles.myImage,'Parent',handles.axes2);
    case 2
        image4=handles.myImage;
        image41(:,:,1)=image4(:,:,1);
        image41(:,:,2)=0;
        image41(:,:,3)=0;
        image(image41, 'Parent', handles.axes2);
    case 3
        image5=handles.myImage;
        image51(:,:,2)=image5(:,:,2);
        image51(:,:,1)=0;
        image51(:,:,3)=0;
        image(image51, 'Parent', handles.axes2);
    case 4
        image6=handles.myImage;
        image61(:,:,3)=image6(:,:,3);
        image61(:,:,1)=0;
        image61(:,:,2)=0;
        image(image61, 'Parent', handles.axes2);
    case 5
        imagebw=im2bw(handles.myImage,0.5);
        imshow(imagebw, 'Parent', handles.axes2)
    case 6
        im_gray=rgb2gray(handles.myImage);
        imshow(im_gray, 'Parent', handles.axes2);

```



```

    otherwise
        im_gray3=imnoise(handles.myImage, 'salt & pepper');
        imshow(im_gray3, 'Parent', handles.axes2);
end

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2 contents
as cell array
%    contents{get(hObject,'Value')} returns selected item from popupmenu2
guidata(hObject, handles);
v=get(handles.popupmenu2,'value');
    im_gray7=rgb2gray(handles.myImage);
    im_gray8=imnoise(im_gray7, 'salt & pepper');

switch v
    case 1
        imshow(handles.myImage,'Parent',handles.axes3);
    case 2
        im_filt=medfilt2(im_gray8);
        imshow(im_filt, 'Parent', handles.axes3);
    case 3
        filtro=1/9*[1 1 1;1 1 1;1 1 1];
        filtro_suavizado=filter2(filtro,double(im_gray8));
        image(filtro_suavizado, 'Parent', handles.axes3);
    case 4
        [a b]=imhist(im_gray8);
        stem(b,a, 'Parent', handles.axes3)

    otherwise

```

```

        im_filt2=medfilt2(im_gray8);
        filtro3=1/9*[1 1 1;1 1 1;1 1 1];
        filtro_suavizado2=filter2(filtro3,double(im_filt2));
        image(filtro_suavizado2, 'Parent', handles.axes3);

end

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
rgb = getimage(handles.axes2);
if isempty(rgb), return, end

formatos = {'*.jpg','JPEG (*.jpg)';*.tif','TIFF (*.tif)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR IMAGEN');
if nomb==0, return, end
fName = fullfile(ruta,nomb);
imwrite(rgb,fName);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
filtro = getimage(handles.axes3);
if isempty(filtro), return, end

formatos = {'*.jpg','JPEG (*.jpg)';*.tif','TIFF (*.tif)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR IMAGEN');
if nomb==0, return, end
fName = fullfile(ruta,nomb);
imwrite(filtro,fName);

```

Anexo B
Dispositivos utilizados



Some Measurements on DVB-T Dongles with E4000 and R820T Tuners: Image Rejection, Internal Signals, Sensitivity, Overload, 1dB Compression, Intermodulation

August 2013
HB9AJG

Contents:

1. Motivation
2. Test Setup
3. Image Rejection
4. Internal Signals
5. Sensitivity
6. Overload and 1dB Compression
7. Intermodulation
8. Aliasing
9. Summary
10. What do we learn from these tests?

1. Motivation

After having read quite a few posts in the user group <http://uk.groups.yahoo.com/group/SDRSharp> about the sensitivity or "deafness" of these dongles, but not having come across any figures, I feel constrained to post the results of my tests. I have measured two types of dongles:

- a Terratec dongle, containing an E4000 tuner and the RTL2832, VID + PID = 0CCD 00D7, designated *E4000* further down.
- a "no name" dongle marked "DVB-T+DAB+FM ", containing an R820T tuner and the RTL2832, VID + PID = 0BDA 2838, designated *R820* further down.

2. Test Setup

Instruments used:

- HF Signal Generators HP8657A (0.1...1040MHz) and HP8640B (0.5...520MHz)
- AF Voltmeter HP333A.

Software used: SDR# v1.0.0.135

Settings:

- *Configure*: 2.048MSPS, RTL AGC off, Tuner AGC off, RF Gain max (49dB for the R820, 42dB for the E4000)
- *Radio*: CW-U, Bandwidth 500Hz, Filter Order 300
- *AGC* off
- *FFT Display*: Window Blackman-Harris, Resolution 262144.
- In the spectrum display the noise floor appears at about -40dBm. In order to display the levels of received signals closer to reality, in *SDRSharp.exe.Config* I modified the value -40.0 in line `<add key="fftOffset" value="-40.0" to "-120.0"`.

Having been an HF engineer and a ham radio operator for over 40 years, I am used to measure receiver sensitivity in dBm. I am aware of those UHF enthusiasts who are used to rate their receivers by Noise Figure (NF). It is easy to go from dBm to NF, however, by using the formula:

$$NF = \text{Power of the MDS (dBm)} + 174\text{dBm} - 10\log(\text{Bandwidth}) - \text{Measured S/N}.$$

Example:

MDS (Minimum Discernible Signal) = Signal giving an (S+N)/N of 3dB (measured with the AF Voltmeter).

$10\log(\text{Bandwidth})$ in my measurements is $10\log(500) = 27\text{dB}$.

For (S+N)/N = 3dB, S is equal to N. Thus, S/N is 0dB.

So, for an MDS of, say, -140dBm: $NF = -140 + 174 - 27 - 0 = 7\text{dB}$.

Remark 1: The dongles have a nominal input impedance of 75 Ohms, whereas my signal generators have output impedances of 50 Ohms. My dBm figures take account of the difference of 1.6dB.

Remark 2: The measurements were carried out on one dongle of each type only. Of course, the results may vary by several dBs between individual dongles.

Remark 3: In such measurements uncertainties of +/-1...2dB are common.

3. Image Rejection

The E4000 is a direct conversion receiver; therefore, a "hump" is clearly visible in the center of the spectrum display. It is mostly due to noise on the power source of the dongle, and could probably be reduced by a well filtered, external power supply (I did not try this). **The R820 first mixes down to an IF of a few MHz,** and, therefore, does not show such a hump.

Both dongles present a peak in the center of the spectrum display of SDR#, which for the *E4000* can be reduced, and for the *R820* can be (almost) eliminated by marking *Correct IQ*.

For the *E4000*, marking *Correct IQ* improves the image rejection by about 30dB. Without *Correct IQ* the image rejection of the *E4000* is a poor 21dB (measured on 435MHz), with *Correct IQ* it reaches a more or less acceptable 50dBs. This means that for signals lower than about -90dBm no image will be visible any more in the spectrum display.

Because of the different technology, the *R820* does not have the same image rejection problem. Therefore, marking *Correct IQ* has no effect (except for the peak in the center of the spectrum display).

4. Internal signals

In digital receivers clock signals, poor shielding (space requirements!) and digital processing produce internal signals actually not present at the antenna input ("birdies" in ham speak). Since the dongles are not shielded, I wrapped them in aluminum foil connected to the PC ground. Figures 1 and 2 show typical examples. Harmonics of the clock frequency are quite strong and clearly visible up to at least 1GHz. Regarding birdies, the *R820* is much cleaner than the *E4000*.

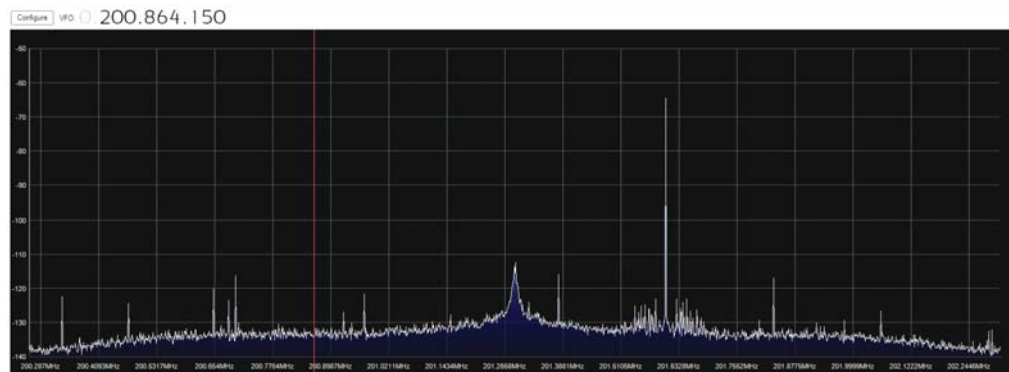


Figure 1 *E4000*: 7th harmonic of the clock frequency 28.8MHz and other "birdies".

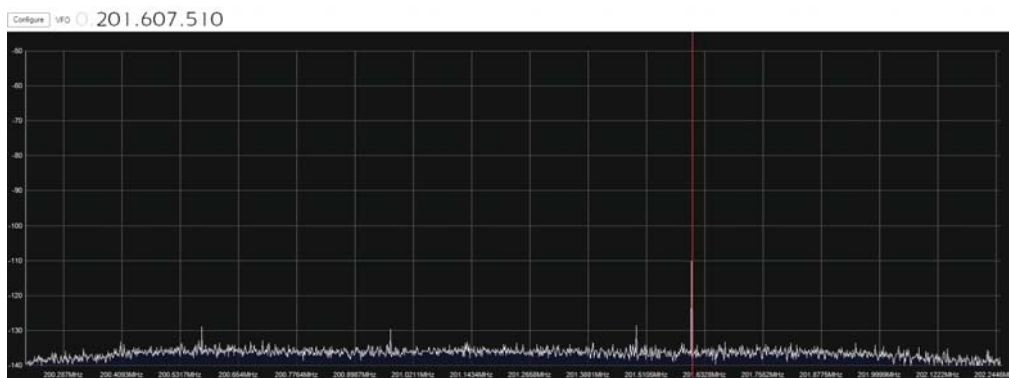


Figure 2 *R820*: 7th harmonic of the clock frequency 28.8MHz and a few, weak birdies. Also note the missing "hump" in the center.

It is easy to recognize birdies: most of them (except the harmonics of the clock, of course) vary their frequency when moving the spectrum window in frequency. Many of them even move up if you move the window down in frequency.

5. Sensitivity

Definition of sensitivity:

MDS (Minimum Discernible Signal) = 3dB (S+N)/N, measured with the HP333A.

My signal generators only go up to 1040MHz, therefore no measurements were possible on higher frequencies. Using harmonics of the generators, I was able to verify that my *E4000* had a range of 51.850...>2080MHz, with a gap of 1105...1268MHz. My *R820* had a range of 24...1766MHz without a gap.

Frequency MHz	Sensitivity <i>E4000</i> dBm	Sensitivity <i>R820</i> dBm
24	n.a.	-127
52	-139	-134
110	-139	-134
145	-141	-134
435	-139	-135
700	-136	-136
1000	-129	-137

Table 1 Sensitivity

Remark: The figures are for signals about in the middle between the left edge and the center of the spectrum display. At the edges the sensitivity is a few dBs less.

Both dongles are very sensitive. Between 50 and about 500MHz the *E4000* is about 5dB more sensitive than the *R820*, but at 1000MHz the *E4000* is about 8dB less sensitive compared to the *R820*.

6. Overload and 1dB Compression

6.1 Overload

Figures 3 and 4 show the effects of a single, strong signal within the window of the spectrum display:

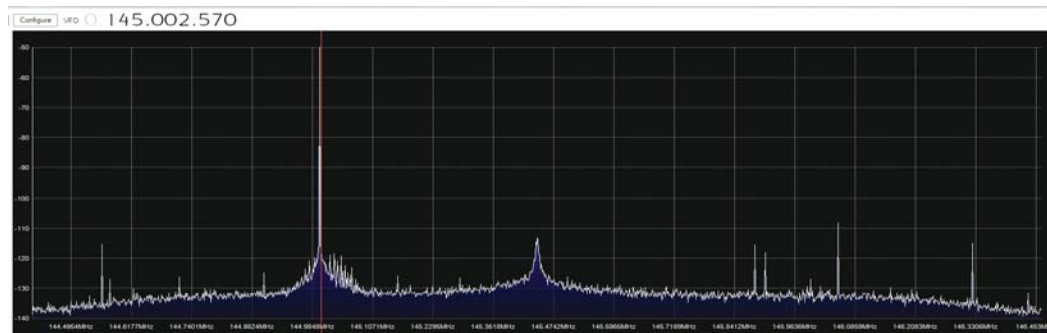


Figure 3 *E4000*: Just before overload at 145MHz/-64dBm (the other signals are "birdies" and aliasing products)

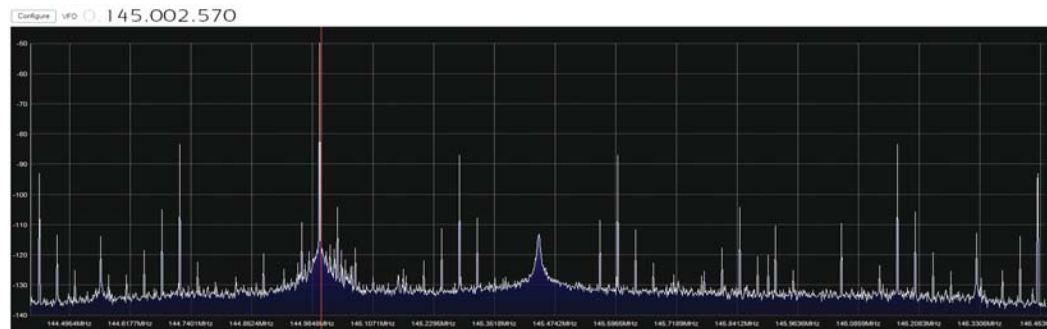


Figure 4 *E4000*: Overload at 145MHz/-63dBm: a multitude of "phantom signals" pop up

The "phantom signals" pop up suddenly, within an increase of the level of the strong signal of only 1dB.

At 145 MHz turning on *RTL AGC* surprisingly degrades the overload limit by about 5dB. Then reducing the gain improves the overload limit, but not by the same amount the gain is reduced. Turning on *Tuner AGC* and maintaining full gain has no effect on the overload limit. And if the gain is then reduced, the overload limit does not improve at all. At 435MHz the behavior is the same.

The effects of overload of the *R820* are similar to Figure 4. At 145MHz turning on *RTL AGC* and maintaining full gain degrades the overload limit by 1...2dBs. Then reducing the gain improves the overload limit by about the same amount the gain is reduced. Turning on *Tuner AGC* is to be avoided: it degrades the overload limit by about 10dBs, independent of the gain setting.

At 435MHz turning on *RTL AGC* and maintaining full gain improves the overload limit by 1...2dBs. Then reducing the gain improves the overload limit by about the same amount the gain is reduced. Turning on *Tuner AGC* is to be avoided: it degrades the overload limit by about 10dBs, independent of the gain setting.

For both dongles it seems there is nothing to be gained from activating *RTL AGC* or *Tuner AGC*.

6.2 1dB Compression

The block diagrams of both dongles show an RF filter after the preamplifier. These filters (and also the subsequent digital signal processing) should help to improve the overload limit as the strong signal is moved away from the center frequency of the spectrum display.

But the level of such a strong input signal cannot be increased without limits, because the preamplifier will saturate at some point (and, ultimately, will give smoke signals), hence reducing the S/N ratio of a desired signal. It is common practice to determine the level of a strong signal that causes a reduction of the S/N ratio of a desired signal by 1dB.

I tried to measure the overload limit and/or the 1dB compression limit for two center frequencies of the spectrum display, 145MHz and 435MHz. The desired signal was close to the center frequency with a level of -80dBm. If overload occurs like in Figure 4, the limit can be determined quite accurately (within 1...2dB). For strong signals further away from the desired signal, the limit is not that well defined anymore, because then the effect of overload becomes weaker, and finally only a handful of "new" signals pop up in the display. In these cases I chose the 1dB compression as the limit. The following figures show overload and/or 1dB compression limits for both dongles, and for center frequencies of 145MHz and 435MHz.

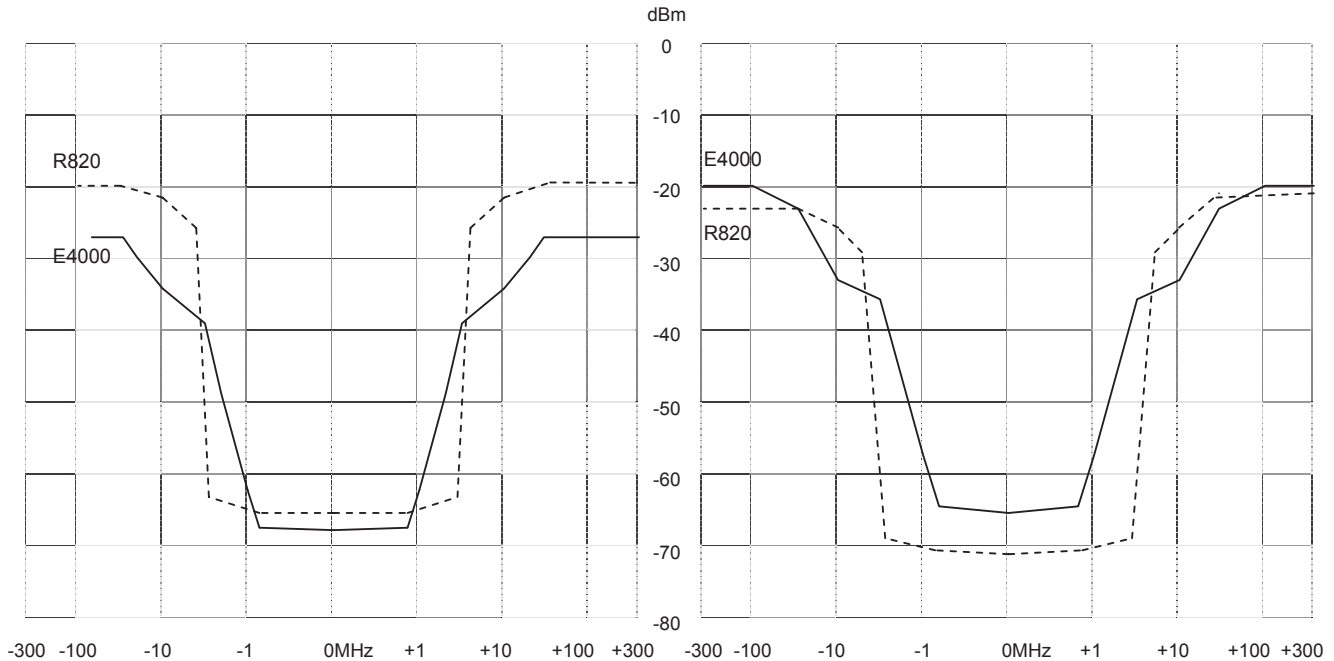


Figure 5 Overload/1dB Compression at 145MHz
Desired signal -80dBm at 145.03MHz
E4000 = ——— R820 = - - - - -

Figure 6 Overload/1dB Compression at 435MHz
Desired signal -80dBm at 435.03MHz
E4000 = ——— R820 = - - - - -

For a weaker desired signal of only -120dBm the measured values differed by only about +/-1dB for both center frequencies and for both dongles. I was very surprised by these results; I had expected the preamplifier to overload at much lower signal levels.

The figures show that the digitally tuned filters (and the subsequent digital signal processing) of both dongles improve the overload/1dB compression limit considerably. The filters are more or less symmetric.

The filter of the E4000 is about +/-0.8MHz wide, but less steep than the filter of the R820. Surprisingly, at 435MHz the filter is a bit steeper than at 145MHz. Within this bandwidth a signal of about -64...-62dBm causes overload. At about +/-5MHz 1dB compression becomes the limiting effect.

The filter of the R820 is about +/-3MHz wide, but steeper than the filter of the E4000. Up to about +/-1MHz the effects of overload caused by a signal of about -62...-68dBm are very similar to those of the E4000. Up to about +/-3MHz the effects of overload are still the same, but getting weaker. At about +/-10MHz 1dB compression becomes the limiting effect.

For the E4000 the overload/1dB compression limit is not linearly dependent of the gain set in Configuration of SDR#: if the gain is reduced by 13/20/30dB, the overload limit is improved by only 7/14/25dB (measured on 145MHz only).

For the R820 the overload/1dB compression limit is nearly linearly dependent of the gain set in Configuration of SDR#: if the gain is reduced by 11/20/30dB, the overload limit is improved by 12/20/30dB (measured at 145MHz only).

7. Intermodulation

If strong signals are fed to a nonlinear amplifier (they all are nonlinear to a certain extent), intermodulation products are generated.

Their general form is: $f_{IM} = +/-n \cdot f_1 +/-m \cdot f_2$ where $n+m$ is called the order of the intermodulation.

Example for Intermodulation of third order (IM3) with $f_1 = 145.1\text{MHz}$ and $f_2 = 145.2\text{MHz}$:

$$f_{IM3} = 2 \cdot 145.1 - 1 \cdot 145.2 = 145.0\text{MHz} \quad \text{and} \quad -1 \cdot 145.1 + 2 \cdot 145.2 = 145.3\text{MHz}$$

and for IM5: $f_{IM5} = 3 \cdot 145.1 - 2 \cdot 145.2 = 144.9\text{MHz}$ and $-2 \cdot 145.1 + 3 \cdot 145.2 = 145.4\text{MHz}$

So, if the two signals are spaced by x kHz, the IM products are appearing every x kHz to the left and to the right of these signals.

IM3 is the strongest. IM levels decrease as their order increases.

7.1 Intermodulation in the E4000

Figure 7 shows intermodulation products generated in the *E4000* by two signals at 145.1 and 145.2MHz. The many signals to both sides of the IM products are due to overload occurring at the same time: The signals at 145.1 and 145.2MHz each have a level of -67dBm. Thus, both signals together exceed the overload level at 145MHz. The highest signal at the far left is the 5th harmonic of the clock, and is always present.

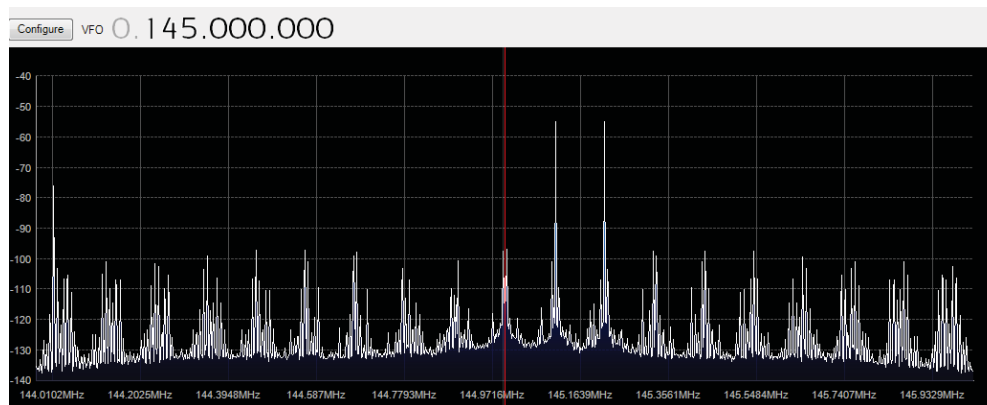


Figure 7 *E4000*, Intermodulation products and overload caused by two signals of -67dBm at 145.1 and 145.2MHz

Figure 8 shows what happens with two strong signals at 147.5 and 147.6MHz, i.e. outside the bandwidth of the filter of the *E4000*. The IM products, evenly spaced at 100kHz are clearly visible. At the same time additional signals pop up, because the two strong signals together reach the overload limit. The same happens if the two strong frequencies are even further away from the center frequency: IM products become visible at about the same level overload occurs.

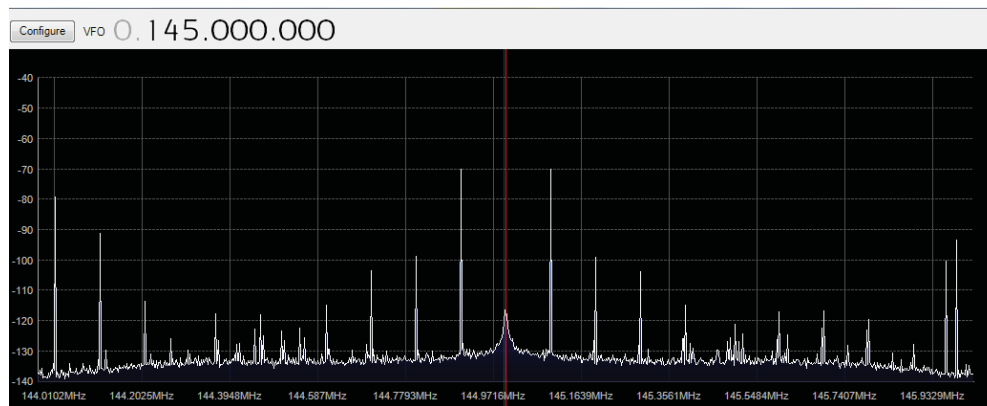


Figure 8 *E4000*, Intermodulation products and overload caused by two signals of -42dBm at 147.5 and 147.6MHz (outside the spectrum display and the filter bandwidth)

Figure 9 shows what happens if the two strong signals are far away (+50MHz) from the center of the spectrum display: IM products begin to appear already at levels of about -40dBm each, i.e. at much lower levels than the overload/1dB compression limit of -24dBm.

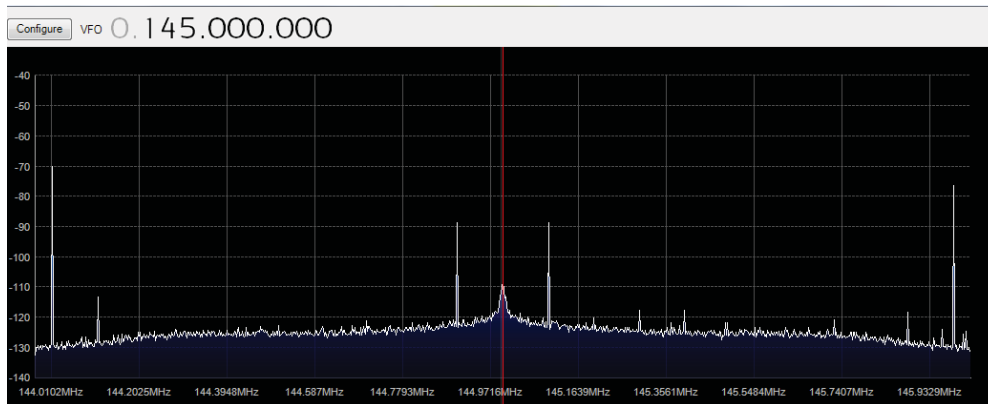


Figure 9 *E4000*, effects of two strong signals of -40dBm at 195.0 and 195.1MHz (far off the center frequency)

At 435MHz the IM in the *E4000* is similar to 145MHz (accounting for the differing overload/1dB compression limits).

7.2 Intermodulation in the *R820*

Figure 10 shows the *R820* with two strong signals of -65dBm each at 145.1 and 145.2MHz. IM products begin to appear, but at the same time both signals together exceed the overload limit. The highest signal at the far left is the 5th harmonic of the clock, and is always present.

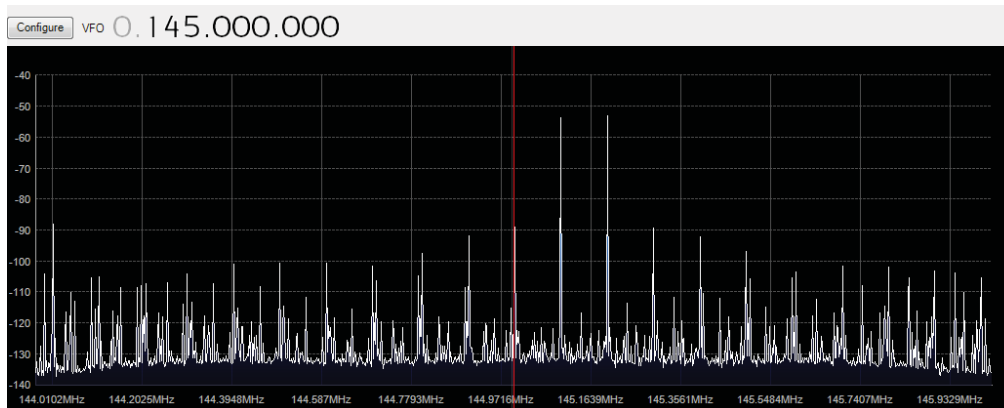


Figure 10 *R820*, intermodulation products and overload caused by two signals of -65dBm at 145.1 and 145.2MHz

Figure 11 shows how the *R820* copes with two strong signals of -37dBm each at 150.0 and 150.1MHz (outside the filter bandwidth): a few IM products begin to appear.

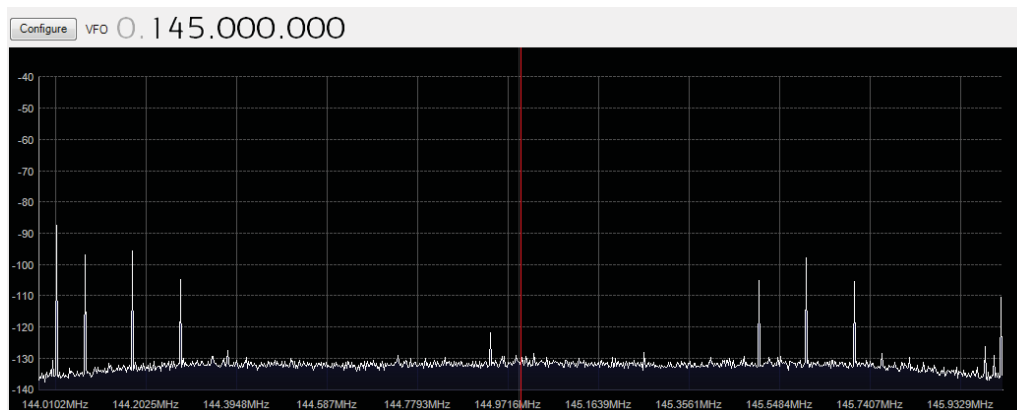


Figure 11 *R820*, effects of two strong signals of -37dBm at 150.0 and 150.1MHz (outside the spectrum display and the filter bandwidth)

Figure 12 shows what happens in the *R820* with two strong signals far away from the center of the spectrum display at the overload/1dB compression limit: the noise floor rises by about 3dB, and a few "new" signals appear.

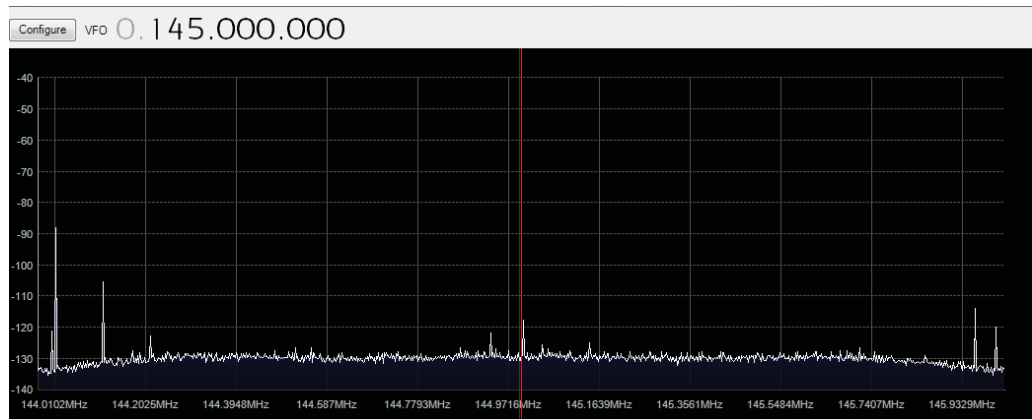


Figure 12 *R820*, effects of two strong signals of -20dBm (at the overload/1dB compression limit) at 195.0 and 195.1MHz (far off the center frequency)

At 435MHz the IM in the *R820* is similar to 145MHz (accounting for the differing overload/1dB compression limits). It seems that for frequencies far off the center frequency the *R820* is less prone to IM than the *E4000*.

8. Aliasing

Aliasing always occurs if an insufficiently band limited signal is sampled, i.e. if the signal to be sampled contains frequencies above half the sampling frequency. Thus, aliasing is an effect showing up in many SDRs, not only in these dongles. In both types of dongles there is not much space for brickwall filters, hence aliasing is to be expected. As an example, Figure 13a shows a desired signal of -80dBm at the far left, and a corresponding aliased signal in the center. If the spectrum display is moved to higher frequencies (by sliding it to the left), the desired signal will disappear at the left, but at the right it will reappear as Alias 2, whereas Alias 1 moves to the right (see Figure 13b). If moving to still higher frequencies, Alias 2 will move more to the left, maintaining its frequency, while its level will decrease because of the effect of the (non-perfect) filtering. At the same time Alias 1 will slide more to the right, maintaining more or less its level, because the filter does not have better wide band attenuation, but increasing in frequency. So, some aliased signals can be recognized by the fact that they are not constant in frequency when the spectrum display is moved. Others are constant in frequency, but not in level. Both types will show the same modulation, e.g. no modulation at the same time.

Aliasing effects are similar in both types of dongles.

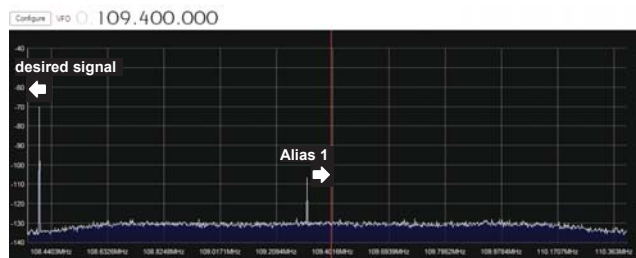


Figure 13b *R820*, Example for aliasing effects: desired signal and an aliased signal

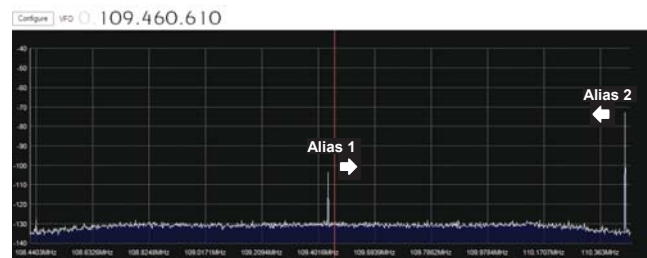


Figure 13b Tuned to a higher frequency, the desired signal disappears at the left and an aliased signal reappears at the right

9. Summary

General notes

- The measurements were carried out on one dongle of each type only. The results may vary by several dBs between individual dongles.
- In such measurements uncertainties of +/-1...2dB are common.

Image Rejection

Because the *E4000* is a Direct Conversion Receiver, it has an Image Rejection problem. By switching on *Correct IQ* in SDR# a more or less acceptable 50dBs are reached. For the same reason, a "hump" shows in the center of the spectrum display. By using a well filtered external power supply (not from the USB connector) the hump might be reduced.

Internal signals

The *E4000* shows many signals actually not present at its input ("birdies"). Birdies are easy to recognize: most of them (except the harmonics of the clock) vary their frequency when moving the spectrum window in frequency. Many of them even move up if you move the window down in frequency.

The *R820* is much cleaner in this respect: besides the harmonics of the clock (28.8MHz) only few birdies show up.

Sensitivity

Both dongles have a very high sensitivity. Between about 50 and 450MHz the *E4000* is about 5dB better than the *R820* (-139dBm vs -134dBm). At 1000MHz the *E4000* is about 8dB less sensitive (-129dBm vs -137dBm). No measurements could be made above 1040MHz.

Overload and 1dB Compression

If a signal is strong enough, it may cause overload, i.e. many (unwanted) signals show up on the spectrum display that are not present at the antenna input. Also, if we listen to a desired signal, another signal (if strong enough) may cause a reduction of the S/N ratio of the desired signal.

Both dongles have a digitally tuned RF filter after the preamplifier that (together with the following digital signal processing) improves the overload/1dB compression limit considerably.

- The filter of the *E4000* is about +/-0.8MHz wide, but less steep than the filter of the *R820*.

- The filter of the *R820* is about +/-3MHz wide, but steeper than the filter of the *E4000*.

For the *E4000* the overload/1dB compression limit is not linearly dependent of the gain set in *Configuration of SDR#*: if the gain is reduced by 13/20/30dB, the overload limit is improved by only 7/14/25dB (measured on 145MHz only).

For the *R820* the overload/1dB compression limit is quite linearly dependent of the gain set in *Configuration of SDR#*: if the gain is reduced by 11/20/30dB, the overload limit is improved by 12/20/30dB (measured at 145MHz only).

For both dongles it seems there is nothing to be gained from activating *RTL AGC* or *Tuner AGC*.

Intermodulation

Intermodulation products in general show up close to the overload/1dB compression limits. However, if the strong signal is on the roll off of the filter, they appear well before this limit.

Aliasing

Aliasing always occurs if an insufficiently band limited signal is sampled, i.e. if the signal to be sampled contains frequencies above half the sampling frequency. Thus, aliasing is an effect showing up in many SDRs, not only in these dongles. In both types of dongles there is not much space for brickwall filters. Therefore, aliasing effects are well visible with both dongles.

10. What do we learn from these tests?

- Both types of dongles are very sensitive. The choice depends on which frequency range you are most interested in.
- Considering birdies and image rejection, the *R820* is much cleaner than the *E4000*.
- Set the spectrum display to show a range of not more than 60db above the noise floor. If a signal is close to the top, you know you are close to overload.
- Both types of dongles are prone to overload by strong signals within their filter bandwidth: +/-0.8MHz for the *E4000*, +/-3MHz for the *R820*. Therefore, keep signals within this bandwidth to not more than about 60dB above the noise floor by reducing the gain. If increasing the gain does not audibly increase the signal to noise ratio of the desired signal any more, reduce the gain by one step. Do not switch on *RTL AGC* or *Tuner AGC*, as it seems there is nothing to be gained.
- Outside their filter bandwidth both types of dongles can live with much higher signals without showing serious degradation. Use the gain control as explained above to check a possible reduction of signal to noise ratio of the desired signal or the appearance of "new" signals not present at the antenna input.
- Intermodulation occurs if several strong signals are present within the bandwidth of the dongle. Their individual voltage levels add up (add 6dB per equally strong signal). Therefore, in frequency bands with many strong signals, e.g. broadcast bands, the gain must be reduced even further. Watch for "new" signals appearing when increasing the gain, and then reduce the gain by one step.
- If very strong signals are present at the antenna input (>-40dBm), they should be attenuated by bandstop or notch filters.

Of course these neat, little toys are no match for "real" software defined receivers. They do have some limitations, but you get a LOT of radio for almost no money! They are fun to experiment with, and to learn more about SDR technologies.

Realtek RTL2832U

The mystery chip at the heart of RTL-SDR.

VERSION 1 ©2015

Corrections are welcome.



1. INTRODUCTION

The [Realtek RTL2832U](#) is a 6 mm square chip (QFN) with 12 pins per side. It uses 3.3 Volts and a 28.8 MHz crystal. The chip's datasheet comes with a non-disclosure agreement. The IC is a DVB-T COFDM Demodulator. DVB-T stands for *Digital Video Broadcasting: Terrestrial* and COFDM stands for *Coded Orthogonal Frequency-Division Multiplexing*. The "U" in the name stands for the USB (2.0) interface. USB uses 4 pins: +Voltage, D-, D+, and ground. A USB extension cord is recommended with RTL-SDR to keep the radio away from computer generated noise. The RTL2832U contains eight general purpose input/output ports and an infrared remote control port.

2. TUNERS

The RTL2832U supports tuners at IF, low-IF, and zero-IF (direct conversion). The most common tuner is the Rafael Micro [R820T](#). This chip has an advantage over the Elonics [E4000](#) in that it only uses two of the RTL2832U's four I/Q pins. The unused pins, if not grounded, can be used as clean inputs for direct sampling purposes. The unused pins are the Q+ and Q- branches.

3. RTL_TEST

In Linux, using the terminal, "[rtl_test](#)" can be used to test the RTL2832U (Ctrl-Z exits). Program switches include: **-s** *sample rate* (the default is 2,048,000 Hz), **-d** *device index*, **-t** *Elonics E4000 tuner*, **-p** *ppm error measurement*, **-b** *output block size*, and **-S** *force synchronous output*.

4. DATA THROUGHPUT

A high-end sound card will push 32-bits at 192-kHz, a data throughput of [6.144 Mbps](#). The RTL2832U has only an 8-bit ADC but can run at 3.2 MS/s (million samples per second, 3200-kHz). This calculates out to a throughput of [25.600 Mbps](#) or four times that of a good sound card. The RTL2832U was made to handle compressed MGEG2 and MPEG4 (H.264) transport streams; including video, audio, and data. Modulation schemes include: 4QAM (QPSK), 16QAM, or 64QAM. In 2k mode (1705 carriers), the carrier spacing is 4464 Hz. In 8k mode (6817 carriers), the carrier spacing is 2232 Hz. Under both modes the sampling rate is 18.284544 MHz and the final LO frequency is a fourth of that or 4.571136 Mhz (low-IF). Common bandwidths are 6, 7, or 8 MHz.

5. IMPORTANT PINOUTS

The important RTL2832U pinouts are as follows: *pins 1/2 In-phase* Input pos/neg, *pins 4/5 Quadrature* Input pos/neg, *pins 11/12* 28.800 MHz *Crystal* Oscillator, *pin 13* Tuner *AGC*, *pins 16/17* SCL/SDA *Tuner Serial* clock/data, *pins 18/19* SCL/SDA 2k (256 byte) *EEPROM Serial* clock/data [S24CS0](#), *pins 25/26* LDO 3.3V [AMS1117](#), *pin 38* *Infrared* Port, and *pins 40/41* USB Data pos/neg.

6. INPUT IMPEDANCE

The input impedance of the I and Q lines of the RTL2832U is “unknown”. However, much is known about the Elonics E4000 chip. It is a “Multi-standard CMOS Terrestrial RF Tuner” in a 5 mm square package (QFN) that has 8 pins per side. It was designed to interface directly to a digital demodulator. The E4000's analog IF outputs; specifically *pin 20* (I+, IVOUTP), *pin 19* (I-, IVOUTN), *pin 18* (Q+, QVOUTP), and *pin 17* (Q-, QVOUTN) are connected, via a capacitor, directly to pins on the RTL2832U. And these I/Q baseband outputs can have a differential peak-to-peak output voltage of 1000 mV. Meaning, the RTL2832U can handle +4 dBm. The E4000 outputs also see an output load of 15k-ohms and 10 pF. Also telling, the outputs' single ended output impedance is 250-ohms (ROUT). A space-wave has an impedance of 377-ohms. Therefore, there may be no significant impedance mismatch between the RTL2832U's I/Q pins and a random wire antenna.

7. ELONICS E4000

The E4000's has a 4.5 dB (or better) noise figure and a 50-ohm RF input. Total gain is ~99 dB, which includes: 30 dB LNA gain, 12 dB mixer gain, and 57 dB IF gain. There are six stages of IF gain, each with a maximum, digitally-programmable gain of: 6 dB, 9 dB, 9 dB, 2 dB, 15 dB, and 15 dB, respectively. The E4000 can handle 2000 mV peak-to-peak or +10 dBm. IIP3 is +5 dBm. It is important to realize that bypassing the E4000 or R820T also bypasses a huge amount of gain. Like radios of old, without RF amplification, use a long wire (SW), large ferrite (MW), or box loop (MW).

8. RTL2832U ARCHITECTURE: DIRECT SAMPLING

The heart of the RTL2832U is its ADC (Analog-to-Digital Converter) and DSP (Digital Signal Processor). It performs Digital Down-Conversion DDC (IF to baseband) via I/Q mixers (phase is 90 degrees apart), digital low-pass filtering, I/Q resampling, and sends 8-bit I/Q data via the USB port.

9. RTL2832U ARCHITECTURE: COFDM

Unused by RTL-SDR is the Fast Fourier Transform FFT unit. This converts time-domain information into frequency-domain information. Time-domain is when the y-axis is amplitude and the x-axis is time. Frequency-domain is when the y-axis is amplitude and the x-axis is frequency. I wrote a FFT algorithm for the [PhiSCAN](#) programs. The output seen on programs like GQRX, SDR#, and SDR Touch is frequency-domain information. Under direct sampling gnuradio does the FFT.

The RTL2832U is a COFDM demodulator, that does things like: symbol synchronization, fine frequency adjustment, phase rotation, channel estimation and correction, inner and outer deinterleaving, Viterbi decoding, RS decoding, forward error correction, adjacent and co-channel interference rejection, pre- and post- and long-echo channel reception, impulse noise cancellation, automatic carrier recovery, channel equalization, channel state information, guard period removal, pilot and TPS decoding, sample rate correction, sample rate interpolation and decimation, AGC delay, measurement of radio frequency levels, SNR estimation, control of the tuner's AGC, MPEG proportional integral derivative filtering, etc. The chip can automatically detect modulation parameters (ex. transmission mode, code rate, guard interval) via patent-pending algorithms.

10. POCKET RADIO HF SDR

See [Pocket Radio HF SDR](#) for details on how to receive MW and SW using no up-converter (~\$43), no fancy metal case (~\$24), no 4:1 balun (~\$11), and no cables (~\$10). DC to 1.7GHz coverage opens up a world of sub-\$20 DX, including: MW, SW, FM, CB, HAM, CORDLESS, RC, NOAA WEATHER, TV, MILITARY, AIR, SHIP, RAIL, TRUNKING SYSTEMS, GPS, SATELLITE, and more.

just_rtfm@<NOSPAM>yahoo.com

[Dr. Phil's Receiver Designs](#)

Copyright 2015

Dr. Phil

the 1990s, the number of people in the UK who are aged 65 and over has increased from 10.5 million to 13.5 million, and the number of people aged 75 and over has increased from 4.5 million to 6.5 million (ONS 2002).

There is a growing awareness of the need to address the needs of older people, and the need to ensure that they are able to live independently in their own homes.

The aim of this paper is to explore the needs of older people, and to discuss the implications for the design of information systems.

The paper is structured as follows. Section 2 discusses the needs of older people, and Section 3 discusses the implications for the design of information systems.

2. Needs

The needs of older people are complex, and they vary from person to person. However, there are some common needs that are shared by many older people.

One of the most important needs of older people is the need to live independently in their own homes. This need is often met by the provision of home care services, such as help with shopping, cooking, and cleaning.

Another important need of older people is the need to be able to access information and services. This need is often met by the provision of information systems, such as those that provide information about local services, or that allow older people to book appointments.

Older people also have a need to be able to communicate with others. This need is often met by the provision of communication systems, such as those that allow older people to talk to their family members, or to their friends.

Finally, older people have a need to be able to participate in social activities. This need is often met by the provision of social services, such as those that provide opportunities for older people to meet and talk to other people.

3. Design

The design of information systems for older people should take account of their needs, and should be designed to be easy to use and to be accessible to all older people.

One of the most important design considerations is the need to ensure that the system is easy to use. This can be achieved by using simple language, and by providing clear instructions and help.

Another important design consideration is the need to ensure that the system is accessible to all older people. This can be achieved by providing alternative ways of accessing the system, such as by providing a large font, or by providing a voice interface.

Finally, the design of information systems for older people should take account of their cognitive abilities. This can be achieved by providing simple and intuitive interfaces, and by avoiding complex tasks and procedures.

4. Conclusion

The needs of older people are complex, and they vary from person to person. However, there are some common needs that are shared by many older people.

The design of information systems for older people should take account of their needs, and should be designed to be easy to use and to be accessible to all older people.

Older people have a need to live independently in their own homes, and to be able to access information and services. They also have a need to be able to communicate with others, and to be able to participate in social activities.

The design of information systems for older people should take account of these needs, and should be designed to be easy to use and to be accessible to all older people.

References

ONS (2002) *Population Statistics*. London: Office for National Statistics.