



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“OSCILOSCOPIO BASADO EN NIOS II”

TESINA DE SEMINARIO

Previa la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Maritza Fernanda Ruiz Betancourt

Genaro Enrique Valverde Spencer

GUAYAQUIL – ECUADOR

AÑO 2013

AGRADECIMIENTO

A Dios por permitirme culminar esta etapa en mi vida, a mis padres, esposo e hijos por el apoyo brindado siempre durante este proceso.

Maritza Ruiz Betancourt

AGRADECIMIENTO

Agradezco a mis padres que supieron enseñarme el camino correcto desde muy pequeño, por su amor y apoyo en todo momento. A mis profesores por sus conocimientos y enseñanzas que ayudaron a convertirme en un profesional, en especial al Ing. Ronald Ponguillo director de esta tesis. A mi compañera de tesis por su esfuerzo y trabajo en equipo para sacar adelante este proyecto. A todas aquellas personas que directa o indirectamente me ayudaron durante mi etapa universitaria. Finalmente agradezco a Dios por todas las bendiciones recibidas.

Genaro Valverde Spencer

DEDICATORIA

A mis padres, esposo e hijos por ser el motor que me ha impulsado a seguir adelante siempre.

Maritza Ruiz Betancourt

Dedico este trabajo a mi esposa y familia, producto de su apoyo y eterna confianza finalizo esta etapa de mi vida.

Genaro Valverde Spencer

TRIBUNAL DE SUSTENTACIÓN

Ing. Ronald Pongullo

PROFESOR DEL SEMINARIO DE GRADUACIÓN

Ing. Víctor Asanza

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Maritza Fernanda Ruiz Betancourt

Genaro Enrique Valverde Spencer

RESUMEN

El presente proyecto consiste en la implementación de un Osciloscopio, mediante el uso del núcleo NIOS II de Altera que permite implementar un computador embebido en un FPGA. Una de las principales características del procesador NIOS II es que su lógica puede ser programada en lenguaje C/C++ mediante el uso del software NIOS II IDE, esto permite que el diseño sea comprendido y mejorado con facilidad. Para el desarrollo del proyecto se utilizó la tarjeta DE 2 de Altera, que posee un FPGA CYCLONE II que es compatible con el sistema NIOS II.

Para la realización del Osciloscopio utilizamos 3 etapas: acoplamiento de la señal de entrada, conversión Analógica Digital (ADC por sus siglas en inglés) y procesamiento digital de la señal (DSP por sus siglas en inglés). En la primera etapa la señal es amplificada y acoplada para su posterior conversión digital en la segunda etapa. Finalmente esta la etapa DSP que es el corazón del sistema y cumple las siguientes funciones: interpretación de las directivas de control en la interface de usuario, control de las etapas de acoplamiento y ADC, procesamiento de la señal digital recibida por la etapa ADC, impresión final en formato VGA.

El proyecto se ha estructurado en 5 capítulos como se lo explica a continuación:

En el **primer capítulo**, se indican los objetivos específicos y generales del proyecto, explicando las limitaciones y alcances para su implementación.

En el **segundo capítulo**, se expone los conceptos básicos de los elementos utilizados para el desarrollo del proyecto, tales como la definición de un osciloscopio, el sistema NIOS II y la tarjeta de desarrollo DE 2 de Altera.

En el **tercer capítulo**, se muestra el diseño del proyecto en hardware y software, para lo cual se mostraran diagramas de flujo para una mayor comprensión del lector.

Para el **cuarto capítulo**, se describe la implementación del proyecto, definiendo los valores de los elementos utilizados, así como los valores finales de las variables utilizadas para la implementación de hardware.

Finalmente en el **quinto capítulo** están plasmadas las pruebas realizadas en el osciloscopio desarrollado, mostrando imágenes y tablas con los resultados obtenidos. Estas pruebas son la base para la definición de los parámetros de operación del sistema, así como de las conclusiones y recomendaciones.

ÍNDICE GENERAL

AGRADECIMIENTO.....	II
AGRADECIMIENTO.....	III
DEDICATORIA.....	IV
TRIBUNAL DE SUSTENTACIÓN.....	V
DECLARACIÓN EXPRESA.....	VI
RESUMEN.....	VII
ÍNDICE GENERAL.....	IX
ABREVIATURAS.....	XIII
ÍNDICE DE FIGURAS.....	XIV
ÍNDICE DE TABLAS.....	XVI
INTRODUCCIÓN.....	XVII
CAPITULO 1.....	1
1. Generalidades.....	1
1.1 Objetivos.....	1

1.1.1	Objetivo General	1
1.2	Alcance y Limitaciones del Proyecto	2
1.3	Justificación	3
CAPITULO 2.....		4
2.	Marco Teórico	4
2.1	Osciloscopio	4
2.2	El FPGA y el procesador NIOS II de Altera	6
2.2.1	El FPGA	6
2.2.2	El procesador NIOS II de Altera	7
2.2.3	Computadora Media DEII	8
2.3	Tarjeta de desarrollo académico DEII	12
2.3.1	Configuración del Software del FPGA.....	13
2.3.2	Pulsadores e Interruptores	14
2.3.3	Entradas de Reloj.....	16
2.3.4	Salida VGA.....	16
CAPITULO 3.....		18
3.	Diseño del sistema.	18

3.1	Adaptación de señal	19
3.2	Muestreo y conversión ADC	25
3.3	Etapa DSP	29
3.3.1	Controlador_Adc	31
3.3.2	Gen_Clk_Adc	37
3.3.3	Rf_Amp:	40
3.3.4	V_Offset_Dig:	43
3.3.5	Nios_System	44
CAPITULO 4.....		56
4.	Implementación del Sistema	56
4.1	Implementación de Hardware	56
4.2	Implementación de Software / Hardware	58
CAPITULO 5.....		61
5.	Pruebas y Resultados	61
5.1	Medición y Corrección de Voltaje Offset del circuito Electrónico.	62
5.2	Variación del voltaje offset digital.....	64
5.3	Medición de factores de Amplificación.....	65

5.4	Medición de frecuencia de muestreo generada	67
5.5	Variación de la frecuencia de la señal de entrada	68
5.6	Parámetros de operación del sistema	73
CONCLUSIONES.....		75
RECOMENDACIONES		77
ANEXO A		80
ANEXO B		89
ANEXO C		93
ANEXO D		100
ANEXO E		104
ANEXO F.....		108
BIBLIOGRAFIA.....		132

ABREVIATURAS

ADC	Analog-to-Digital Converter
ASIC	Application Specific Integrated Circuit
DE2	Development and Education Board
DSP	Procesamiento Digital de Señales
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
HDL	Hardware Description Language
JTAG	Joint Test Action Group
SDRAM	Synchronous Dynamic Random Access Memory
SOPC	System on a Programmable Chip
UART	Universal Asynchronous Receiver-Transmitter
VGA	Video Graphics Adapter
VHDL	Very High Description Language

ÍNDICE DE FIGURAS

Fig. 2. 1	Diagrama de bloques de computadora media DEII.....	9
Fig. 2. 2	Coordenadas del buffer de pixeles en la pantalla VGA.....	11
Fig. 2. 3	Coordenadas del buffer de caracteres en la pantalla VGA	12
Fig. 2.4	Diagrama de bloques de la tarjeta DEII de Altera.....	13
Fig. 2. 5	Esquemático de la salida VGA en la tarjeta DEII de Altera.....	17
Fig. 3. 1	Diagrama General del Sistema	18
Fig. 3. 2	Diagrama de Bloques del Circuito Adaptador de Señal	19
Fig. 3. 3	Circuito Seguidor.....	21
Fig. 3. 4	Desplazamiento Vertical.....	21
Fig. 3. 5	Diagrama de Resistencias	22
Fig. 3. 6	Amplificación RF.	22
Fig. 3. 7	Esquemático del Multiplexor CD4067 BE.....	23
Fig. 3. 8	Amplificación Final de la Señal	24
Fig. 3. 9	Diagrama General de la Conversión Analógica - Digital	26
Fig. 3. 10	Diagrama de tiempo de ADC	28
Fig. 3. 11	Diagrama de Bloques de la etapa DSP.....	30

Fig. 3. 12	Diagrama ASM del Controlador ADC	32
Fig. 3. 13	Diagrama ASM del Generador del reloj.....	38
Fig. 3. 14	Diagrama ASM del Bloque de Amplificación	42
Fig. 3. 15	Diagrama de Flujo del desarrollo de software	46
Fig. 3. 16	Diagrama de Flujo de la captura del Símbolo	48
Fig. 3. 17	Diagrama de Borrado del símbolo anterior.....	50
Fig. 3. 18	Creación de Cuadrícula.....	52
Fig. 3. 19	Diagrama para impresión del símbolo actual	53
Fig. 4. 1	Implementación de circuito analógico	57
Fig. 5. 1	Esquema de pruebas de voltaje offset	62
Fig. 5. 2	Pruebas de Variación de frecuencia de la señal de entrada	69
Fig. 5. 3	Graficas de Salida: (a) Frecuencia de muestreo 2.7 veces la frecuencia de la señal. (b) Frecuencia de muestreo 7.8 veces la frecuencia de la señal.....	71
Fig. 5. 4	Graficas de Salida: (a) Frecuencia de muestreo 138 veces la frecuencia de la señal. (b) Frecuencia de muestreo 7800 veces la frecuencia de la señal.....	72

ÍNDICE DE TABLAS

Tabla I	Mapeo de puertos del FPGA con los pulsadores e interruptores en la tarjeta DEII	15
Tabla II	Mapeo de puertos del FPGA con las señales de reloj en la tarjeta DEII	16
Tabla III	Descripción de la Señales de la Etapa de Adaptación.....	20
Tabla IV	Amplificaciones esperadas en condiciones ideales	23
Tabla V	Parámetros de control de ADC	25
Tabla VI	Valores de frecuencia de muestreo válidos para $N_{sp}=1000$, $N_{red}=25$	40
Tabla VII	Valores de elementos utilizados.....	58
Tabla VIII	Definición de parámetros utilizados en el software	60
Tabla IX	Pruebas de Calibración Offset	64
Tabla X	Valores medidos en la Amplificación	66
Tabla XI	Tabla de Frecuencias Obtenidas por el sistema	68
Tabla XII	Parámetros finales del sistema	73

INTRODUCCIÓN

En la actualidad los continuos avances en el campo investigativo a nivel electrónico, ha permitido el desarrollo de FPGA's, implementados a partir de la interconexión de bloques de lógica programable. La gran ventaja que presentan estos sistemas es que permiten al usuario una mayor versatilidad en el diseño, es decir que puede ser alterado simplemente cambiando la programación del hardware utilizando lenguaje HDL(Lenguaje de descripción de hardware), lo que con los circuitos electrónicos convencionales sería un proceso muy complicado, es por ello que han sido utilizados en una gran cantidad de soluciones tales como DSP(Procesamiento Digital de Señales), sistemas aeroespaciales, bioinformática, sistemas de imágenes para medicina entre otras.

Esta gran flexibilidad es una de las claves del éxito de los FPGAs, los cuales están en constante evolución y en la actualidad presentan características lo suficientemente robustas para la implementación de sistemas en un Chip (SOPC por sus siglas en Inglés) como procesadores y computadoras embebidas.

Teniendo en mente fomentar el desarrollo académico local en esta área, en nuestro proyecto se implementará un osciloscopio mediante el procesador

embebido NIOS II de Altera. Es por ello que se ha decidido la utilización de la tarjeta de desarrollo DE2 de Altera, la cual posee un FPGA compatible con el sistema NIOS II.

CAPITULO 1

1. GENERALIDADES

En este capítulo daremos a conocer los objetivos planteados para el presente proyecto así como sus alcances y limitaciones.

1.1 Objetivos

1.1.1 Objetivo General

Construir un osciloscopio basado en el Microprocesador NIOS II con visualización en un monitor VGA

1.1.2 Objetivos Específicos

- Adquirir conocimientos de tarjeta DE2 de Altera.
- Entender el funcionamiento de un osciloscopio
- Desarrollar hardware y software que sirvan para la

- implementación del osciloscopio

1.2 Alcance y Limitaciones del Proyecto

Entre los alcances del proyecto se tiene:

- ✓ Desarrollar un sistema que permita tomar señales eléctricas de entrada y mostrarlas en un monitor VGA.
- ✓ Se contará con un solo canal para la muestra de las imágenes en la pantalla, además se podrá variar la frecuencia de muestreo de la señal.
- ✓ Para la etapa de adaptación de señal y ADC se implementará un circuito electrónico que permita una amplificación variable y la inserción de voltaje offset para el desplazamiento vertical. El voltaje de entrada del circuito electrónico debe estar en un rango de -5 a 5 voltios.
- ✓ Para la etapa DSP Se utilizara la tarjeta DE 2 de altera que posee un FPGA CYCLONE II que es compatible con el sistema NIOS II.
- ✓ Se utilizara el software Quartus II para la programación HDL, y NIOS II IDE para la programación de la lógica del procesador NIOS II en C/C++.

Entre las limitaciones se tienen las siguientes:

- ✓ La baja disponibilidad y variedad de elementos electrónicos en el mercado local.
- ✓ La frecuencia máxima de muestreo estará limitada a las características del ADC que se utilice.
- ✓ La resolución de la gráfica del osciloscopio estará limitada a la resolución del procesamiento digital de la salida VGA en el procesador NIOS II.

1.3 Justificación

El continuo avance en el desarrollo de FPGA's y la poca investigación llevada a cabo en este campo en nuestro país ha sido una de las principales razones para llevar a adelante este proyecto de realizar un osciloscopio utilizando el microprocesador NIOS II, ya que con ello estaremos motivando la investigación y el desarrollo de aplicaciones de parte de los estudiantes, haciendo uso de esta herramienta de gran utilidad y con grandes ventajas no solo en el ámbito académico.

CAPITULO 2

2. Marco Teórico

En el presente capítulo exploramos los fundamentos teóricos de nuestro trabajo, para esto hemos dividido el capítulo en tres secciones. En la primera sección exponemos las características básicas de un osciloscopio. En la segunda sección cubrimos los conceptos del FPGA y de manera más concreta del procesador NIOS II provisto por ALTERA que permite la implementación de una computadora embebida en un FPGA. Finalmente, en la tercera sección se exponen los conceptos básicos de la tarjeta DE2 enfocándonos en los componentes que se utilizarán en nuestra implementación.

2.1 Osciloscopio

El Osciloscopio es el instrumento habitual para visualizar, medir, analizar y registrar, mediante un trazo no permanente, tensiones variables en el tiempo. A pesar de su extenso uso, el osciloscopio es relativamente desconocido, pues a menudo se

considera que ofrece una visión “continua” de los valores instantáneos de la señal, cuando en realidad solo representa una superposición de fragmentos o una sucesión de valores instantáneos de la misma [1].

En términos de implementación los osciloscopios pueden dividirse en analógicos y digitales. Los osciloscopios analógicos son implementados mediante un tubo de rayos catódicos y con el tiempo han sido reemplazados casi en su totalidad por los osciloscopios digitales. De hecho, la única ventaja con su contraparte digital es su bajo costo ya que el procesamiento digital es mucho más eficiente y no acarrea la fragilidad que implica un tubo de rayos catódicos [2].

Por otro lado, el osciloscopio digital está conformado básicamente por un circuito de muestreo y conversión analógica digital, una memoria digital y una conversión digital analógica para su posterior impresión en pantalla. Entre las ventajas que tiene un procesamiento digital con su contraparte analógica tenemos [3]:

- Capacidad de guardar y sostener eventos en “un disparo”.
- Mejor captura y visualización de eventos transientes.

- Potencialidad de imprimir la forma de onda y otro tipo de datos.
- Una mejor visualización en pantalla de la señal y los parámetros de configuración del osciloscopio.
- Capacidad de funcionar como un voltímetro digital y medidor de frecuencia.

2.2 El FPGA y el procesador NIOS II de Altera

2.2.1 El FPGA

Un FPGA (del inglés Field Programmable Gate Array) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada dinámicamente mediante un lenguaje de descripción de hardware. La diferencia entre los FPGA y otros dispositivos de lógica programable (como los ASICs) radica en dos aspectos. Primero, las puertas lógicas no se encuentran implementadas de manera física, sino que para implementar la lógica deseada se utilizan generadores de funciones y memorias discretas. Segundo, los FPGAs utilizan celdas de RAM estáticas para mantener la configuración. Estas dos características

son las que permiten que un dispositivo sea configurado o inclusive reconfigurado una vez que ya está en producción [4] [5].

Esta gran flexibilidad es una de las claves del éxito de los FPGAs, los cuales están en constante evolución y en la actualidad presentan características lo suficientemente robustas para la implementación de sistemas en un Chip (SoC por sus siglas en Inglés) como procesadores y computadoras embebidas.

2.2.2 El procesador NIOS II de Altera

EL Procesador Nios II es un núcleo de Altera que permite la implementación de un procesador de propósito general RISC (del inglés Reduced Instruction Set Computer, en español Computador con Conjunto de Instrucciones Reducidas) embebido en un FPGA.

En otras palabras un procesador Nios II es equivalente a un microcontrolador o “computadora en un chip” que incluye un procesador y una combinación de periféricos y memoria en un solo circuito integrado. Como cualquier

familia de microcontroladores, todo sistema de procesador NIOS II utiliza un conjunto de instrucciones consistentes y un modelo de programación. Una de las características más relevantes de este procesador es que el desarrollo de software es basado en lenguaje C/C++ mediante el uso de las herramientas de construcción de software (SBT por sus siglas en inglés) NIOS II para Eclipse[6].

2.2.3 Computadora Media DEII [7]

El sistema denominado computadora media DEII está orientado a ser una plataforma para experimentos de sistemas embebidos y computacionales. En resumidas cuentas se trata de un ejemplo de la implementación de un sistema NIOS II orientado al uso de la tarjeta de desarrollo académico DEII de altera y puede servir como punto de partida para cualquier desarrollo utilizado con dicho procesador. Para esto, a parte del procesador NIOS II posee otros componentes como memorias, dispositivo de audio y video y periféricos de entrada/salida, todos estos implementados por núcleos

de Altera que se pueden descargar desde la página web de Altera.

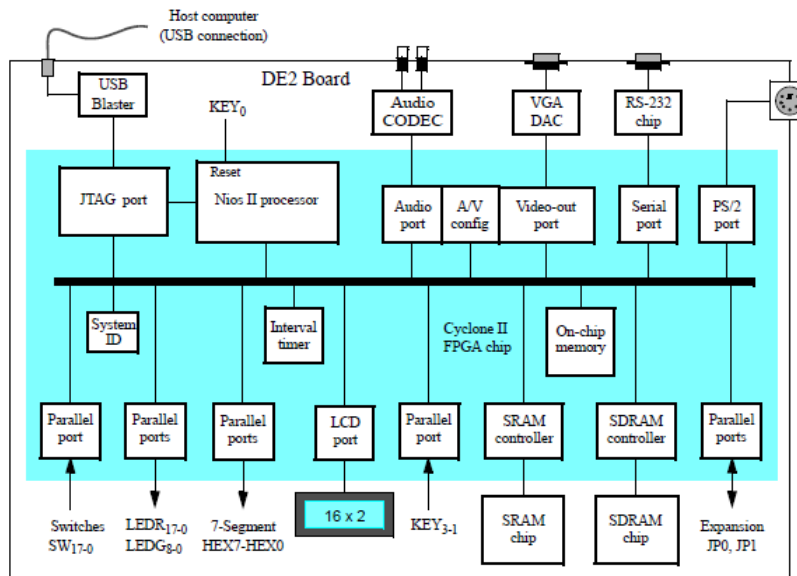


Fig. 2. 1 Diagrama de bloques de computadora media DEII.

Si tenemos en cuenta que la implementación de un osciloscopio a nivel DSP consiste básicamente en la impresión en pantalla de la señal y sus parámetros, la utilización del puerto de salida de video de la computadora media DEII puede ser un componente básico que simplificaría en gran parte el desarrollo.

El puerto de salida de video de la computadora DEII es básicamente un controlador VGA que permite la conexión del sistema a un monitor VGA y soporta una resolución de hasta 640x480 pixeles. La imagen mostrada por el controlador VGA proviene de dos fuentes: el buffer de pixeles y el buffer de caracteres.

El buffer de píxeles envía valores de píxeles con colores desde una memoria de almacenamiento hacia el controlador VGA para su visualización en pantalla. En las condiciones originales de la computadora DEII esta memoria es manejada por una SRAM externa de acuerdo al modelo de fig. 1. La resolución del buffer de pixeles es de 320x240 por lo cual cada valor del pixel es replicado en las coordenadas (x, y) ya que el controlador tiene el doble de resolución. Otro aspecto a considerar es que el origen del plano de coordenadas está situado en la parte superior izquierda de la pantalla y el sentido del eje y es negativo. A nivel de software el buffer de pixeles es manejado por la librería `altera_up_avalon_pixel_buffer.h` y debe ser instanciada en el código C, para una mayor

referencia de las diferentes funciones de dicha librería ver [8]

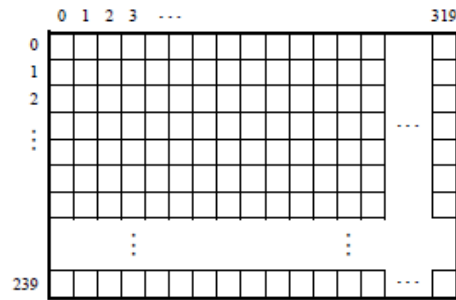


Fig. 2. 2 Coordenadas del buffer de píxeles en la pantalla VGA

Por otro lado, el buffer de caracteres se encarga de mapear el código ASCII de un carácter en un conjunto de píxeles para su visualización en pantalla. Para esto utiliza una memoria interna del computador DEII (memoria On-Chip en fig. 2.1). El buffer provee una resolución de 80x60 caracteres por lo cual cada carácter ocupa un bloque de 8x8 píxeles. Al igual que el buffer de píxeles, el origen del plano de coordenadas está situado en la parte superior izquierda de la pantalla y el sentido del eje y es negativo.

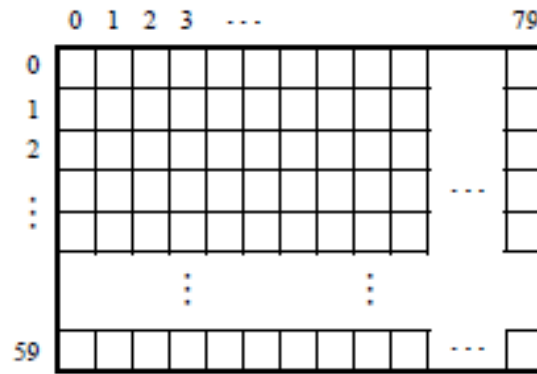


Fig. 2.3 Coordenadas del buffer de caracteres en la pantalla VGA

2.3 Tarjeta de desarrollo académico DEII [9]

La parte central de la tarjeta es el FPGA Cyclone II 2C35 de Altera, el cual es compatible con los núcleos que conforman la solución NIOS II de computador embebido de Altera. El diagrama de bloque de la figura 1.4 detalla los diferentes componentes de la tarjeta. A continuación expondremos el mecanismo utilizado para configurar el FPGA y los principales componentes utilizados en nuestro proyecto.

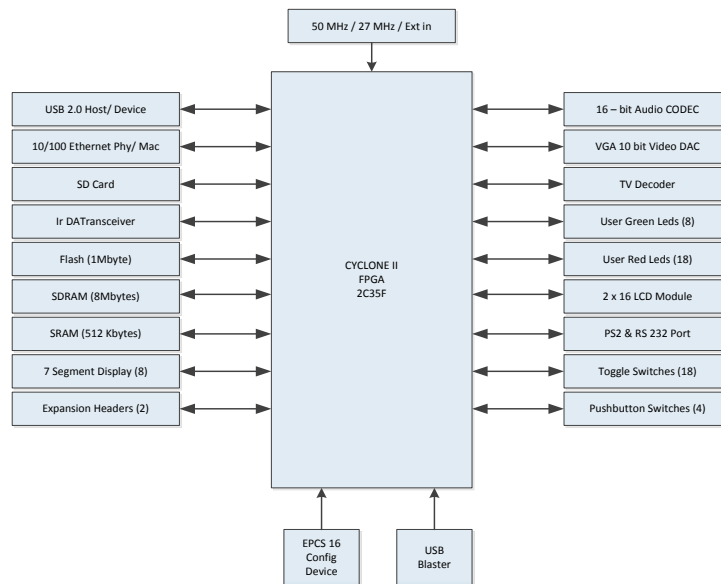


Fig. 2.4 Diagrama de bloques de la tarjeta DE11 de Altera.

2.3.1 Configuración del Software del FPGA

Para la configuración del software del FPGA la tarjeta DE2 posee un chip EEPROM serial que permite cargar los datos de configuración del FPGA cada vez que se encienda la tarjeta. Así, mediante el Software Quartus II de Altera es posible cambiar la configuración de la EEPROM para que el FPGA asuma el comportamiento DSP que se requiera. De hecho, el Quartus II permite realizar los cambios directamente al FPGA, aunque en

esta opción la configuración se perderá una vez que se apague la tarjeta. En definitiva, existen dos métodos de configuración:

- **Programación JTAG:** La configuración es cargada directamente en el FPGA, por lo cual se pierde cuando la energía es desconectada.
- **Programación AS (Serial Activa):** La configuración es cargada en la memoria EEPROM, por lo cual los cambios son permanentes.

2.3.2 Pulsadores e Interruptores

La tarjeta DE2 posee 4 pulsadores y 18 interruptores que están directamente conectados al FPGA. En lo referente a la lógica cuando los pulsadores no están presionados representan un alto (3.3 V), mientras que si se los pulsa representan un bajo (0V). Por otro lado, cuando los interruptores están en posición apagado (más cerca del borde de la tarjeta) representan un bajo (0V) y cuando están en posición encendido un alto (3.3V) lógico. La Tabla I muestra el mapeo de los interruptores y pulsadores con los puertos del FPGA.

Tabla I Mapeo de puertos del FPGA con los pulsadores e interruptores en la tarjeta DEII

Descripción	Señal	Pin del FPGA
Interruptor 0	SW[0]	PIN_N25
Interruptor 1	SW[1]	PIN_N26
Interruptor 2	SW[2]	PIN_P25
Interruptor 3	SW[3]	PIN_AE14
Interruptor 4	SW[4]	PIN_AF14
Interruptor 5	SW[5]	PIN_AD13
Interruptor 6	SW[6]	PIN_AC13
Interruptor 7	SW[7]	PIN_C13
Interruptor 8	SW[8]	PIN_B13
Interruptor 9	SW[9]	PIN_A13
Interruptor 10	SW[10]	PIN_N1
Interruptor 11	SW[11]	PIN_P1
Interruptor 12	SW[12]	PIN_P2
Interruptor 13	SW[13]	PIN_T7
Interruptor 14	SW[14]	PIN_U3
Interruptor 15	SW[15]	PIN_U4
Interruptor 16	SW[16]	PIN_V1
Interruptor 17	SW[17]	PIN_V2
Pulsador 0	KEY[0]	PIN_G26
Pulsador 1	KEY[1]	PIN_N23
Pulsador 2	KEY[2]	PIN_P23
Pulsador 3	KEY[3]	PIN_W26

2.3.3 Entradas de Reloj

La tarjeta DE2 posee 2 osciladores que generan dos señales de reloj de 27MHz y 50MHz respectivamente, además incluye un conector SMA que permite conectar una señal de reloj externa directamente al FPGA. La Tabla II muestra el mapeo de las señales de reloj con los puertos del FPGA.

Tabla II Mapeo de puertos del FPGA con las señales de reloj en la tarjeta DEII

Descripcion	Señal	Pin del FPGA
Reloj de 27Mhz	CLOCK_27	PIN_D13
Reloj de 50MHz	CLOCK_50	PIN_N2
Reloj externo	EXT_CLOCK	PIN_P26

2.3.4 Salida VGA

La tarjeta DE2 posee un conector D-SUB de 16 pines para la salida VGA. Las señales de sincronización VGA son provistas directamente desde el FPGA. Por otro lado, las señales de datos analógicos (rojo, verde y azul

[10]) son construidas mediante el convertidor digital analógico ADV7123 a partir de tres señales digitales de 10 bits provistas desde el FPGA.

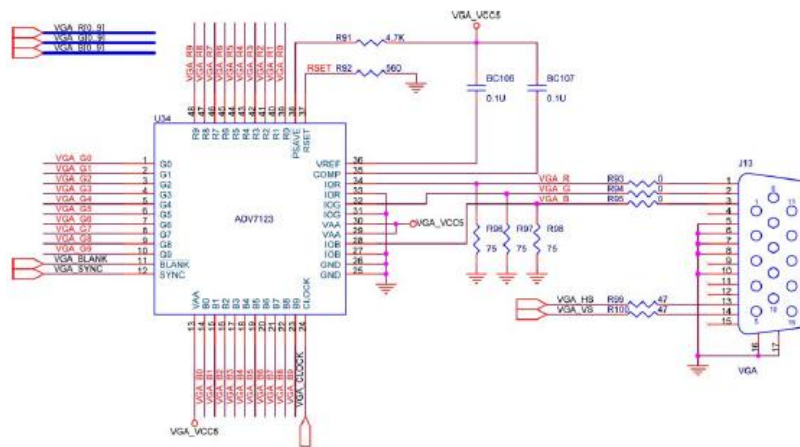


Fig. 2.5 Esquemático de la salida VGA en la tarjeta DEII de Altera.

CAPITULO 3

3. Diseño del sistema.

El diseño del osciloscopio se divide en 3 etapas: adaptación de la señal, muestreo y conversión analógica/Digital (ADC) y procesamiento digital de señales (DSP, por sus siglas en inglés). En el transcurso del presente capítulo se expondrá de manera detallada el diseño de cada uno de estos bloques.

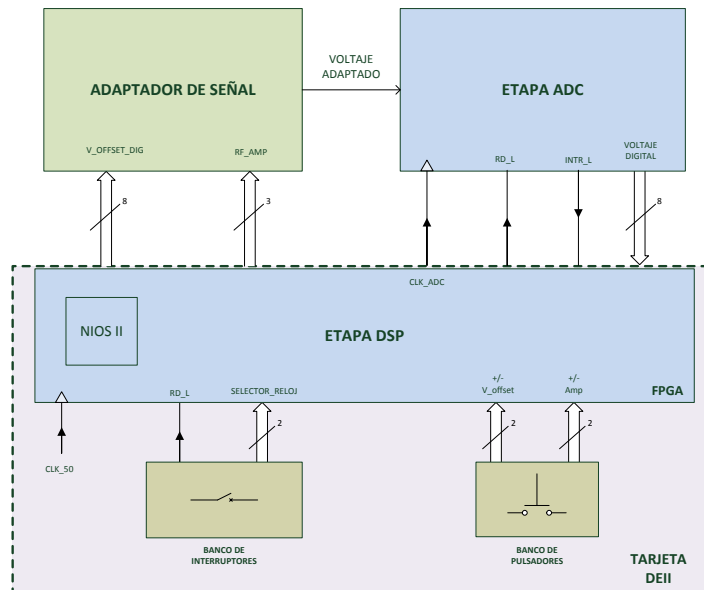


Fig. 3.1 Diagrama General del Sistema

3.1 Adaptación de señal

El objetivo de esta etapa es adaptar la señal de entrada para su correcta interpretación en las etapas ADC y DSP. Como se verá en la sección 3.2, el circuito integrado de la etapa ADC soporta un rango de voltajes entre 0 y 5V, por esta razón es de suma importancia que la señal de entrada sea adaptada a estas condiciones.

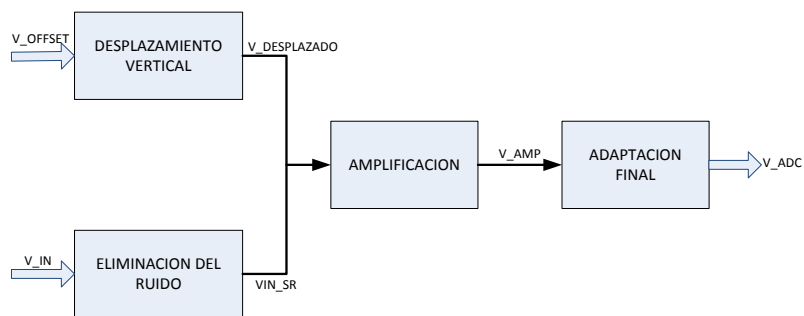


Fig. 3. 2 Diagrama de Bloques del Circuito Adaptador de Señal

Para lograr esto se ha tomado como referencia el diseño electrónico de [11] donde se desarrolla un proyecto similar al nuestro. El circuito básicamente se divide en 4 fases: eliminación de ruido, desplazamiento vertical, amplificación y adaptación final. La diferencia más relevante de nuestro diseño con el de [11] es que, en nuestro caso el control del adaptador

es realizado por la etapa DSP mediante el uso de las señales V_OFFSET_DIG y RF_AMP.

Tabla III Descripción de la Señales de la Etapa de Adaptación.

Nombre	Tipo	Dirección	Descripción
V_IN	Analógica	Entrada	Voltaje de entrada del osciloscopio. La señal debe estar entre -5 y 5V.
V_ADC	Analógica	Salida	Voltaje de salida del bloque adaptador
V_OFFSET_DIG	Digital (8 bits)	Entrada	Señal de control para el desplazamiento vertical del voltaje.
RF_AMP	Digital (3 bits)	Entrada	Señal de control para el factor de amplificación.

La fase de eliminación de ruido es realizada mediante un circuito seguidor de voltaje, construido con un opamp. Este circuito hace que nuestro sistema tenga una alta impedancia de entrada, eliminando el ruido de la señal de entrada.

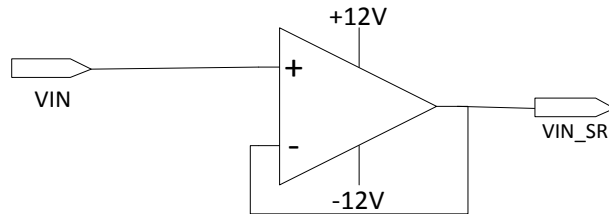


Fig. 3.3 Circuito Seguidor

En la segunda fase, el desplazamiento vertical es controlado mediante la señal digital V_OFFSET_DIG. Dicha señal es convertida a un voltaje analógico mediante una red de resistencias y sumada mediante un circuito de opamps.

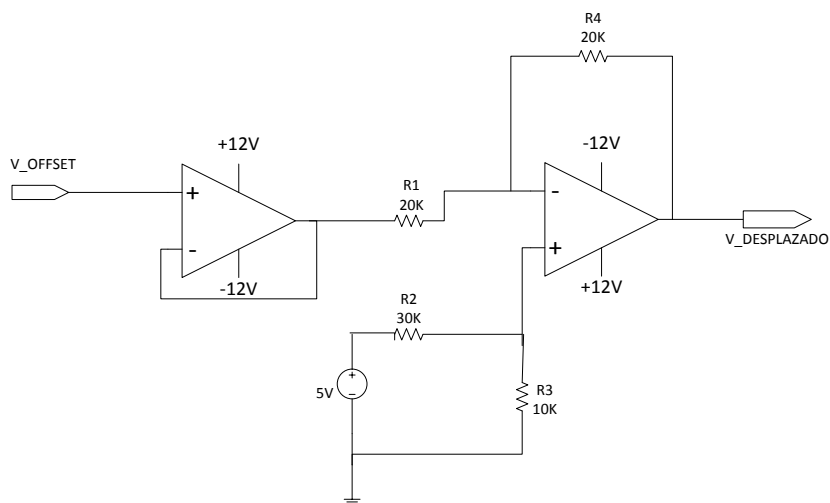


Fig. 3.4 Desplazamiento Vertical.

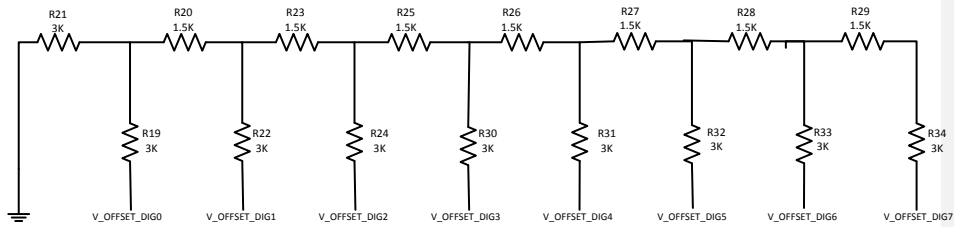


Fig. 3.5 Diagrama de Resistencias

La fase amplificadora de señal se realiza mediante un multiplexor y un circuito sumador. El multiplexor se encarga de seleccionar el factor de amplificación de la señal mediante la selección de una resistencia R_f , en función de la señal de digital de control RF_AMP .

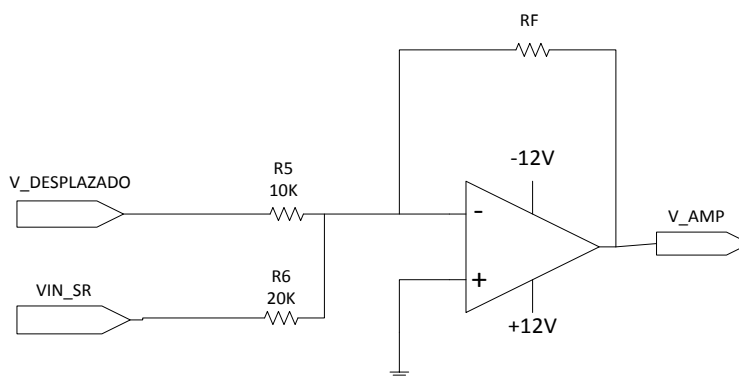
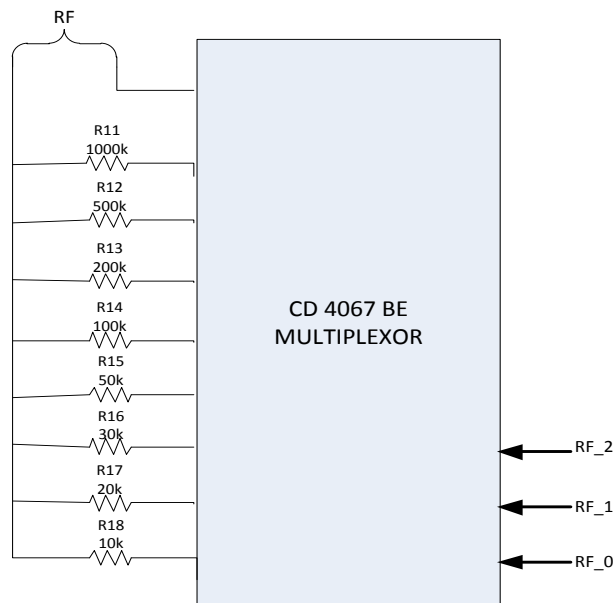


Fig. 3.6 Amplificación RF.

Tabla IV Amplificaciones esperadas en condiciones ideales

MSB		LSB	RESISTENCIA (K Ω)	FACTOR DE AMPLIFICACION
0	0	0	10	1
0	0	1	20	2
0	1	0	30	3
0	1	1	50	5
1	0	0	100	10
1	0	1	200	20
1	1	0	500	50
1	1	1	1000	100

**Fig. 3.7** Esquemático del Multiplexor CD4067 BE

Por último, en la etapa de adaptación final la señal es corregida de polaridad y desplazada a nivel DC para que el ADC reciba valores de voltaje en el rango adecuado.

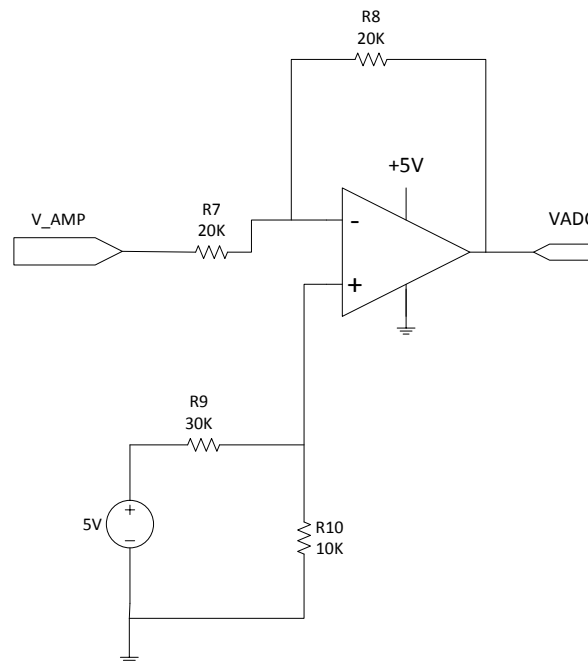


Fig. 3. 8 Amplificación Final de la Señal

Como resultado final, se puede demostrar que el voltaje de salida varía en función del voltaje de entrada, el desplazamiento vertical analógico (V_{OFFSET}) y de un factor de amplificación A , que a su vez depende de la resistencia R_f seleccionada con la señal digital RF_AMP :

$$V_{ADC} = A \cdot \left(\frac{V_{IN}}{2} + V_{OFFSET} \right) + 2.5 \quad (3.1)$$

$$A = 2 \cdot \frac{R_f}{R_6} \quad (3.2)$$

3.2 Muestreo y conversión ADC

Para la implementación de este bloque se ha utilizado el CI ADC0804, debido a su disponibilidad en el mercado local. De acuerdo a las especificaciones del CI [12] los parámetros de control son los siguientes:

Tabla V Parámetros de control de ADC

Parámetro	Dirección	Descripción
CS	Entrada	Señal habilitadora para el funcionamiento del ADC, siempre está en 0, es decir esperando por datos.
RD_L	Entradas	Permite leer los datos convertidos
WR_L	entrada	Utilizada para el inicio de la conversión.
INTR_L	Salida	Señal que activa el ADC una vez que ha

		finalizado el procesamiento de los datos.
VIN	Entrada	0 a 5 V

El esquema de implementación escogido es el de solo lectura, debido a que es el que más se apega a nuestras necesidades. Las señales de control necesarias para su manejo son provistas por la etapa DSP, tal como lo muestra la Tabla V.

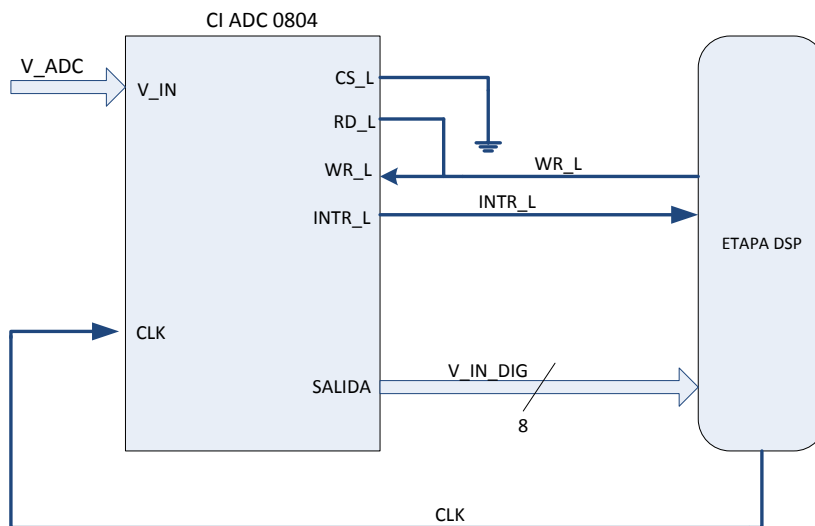


Fig. 3.9 Diagrama General de la Conversión Analógica - Digital

En este esquema el ADC procesa una señal analógica cuando se detecte un voltaje bajo en la entrada RD_L. De acuerdo a [12] el tiempo de la conversión digital T_c está dado por:

$$T_c = 66 \left(\frac{1}{CLK_{ADC}} \right) \quad (3.3)$$

Por otro lado, para garantizar un funcionamiento adecuado del CI la frecuencia de reloj no debe superar los 640kHz, es decir:

$$CLK_{ADC} < 640KHz \quad (3.4)$$

Una vez transcurrido el tiempo de conversión T_c , el ADC activa la señal INTR_L para notificarle a la etapa DSP que la conversión ha finalizado. Es importante tener en cuenta que los datos aún no están disponibles para su lectura, sino que se encuentran en un buffer interno del ADC. Por esta razón, antes de activar la señal RD_L deben transcurrir un tiempo de espera T_{esp} mínimo de 8 flancos de reloj, es decir:

$$T_{esp} > 8 \left(\frac{1}{CLK_{ADC}} \right) \quad (3.5)$$

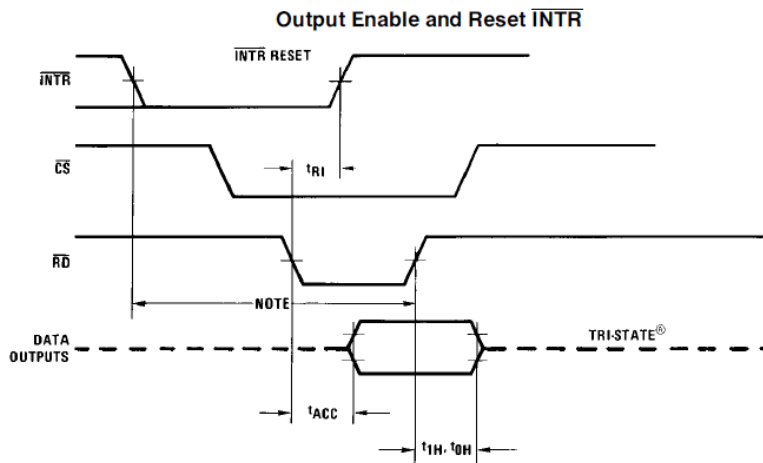


Fig. 3.10 Diagrama de tiempo de ADC [12]

Luego, una vez activada la señal RD_L los datos estarán listos en un tiempo t_{cap} de 200ns. Este dato es importante porque el tiempo de captura (posterior a la activación de RD_L) debe ser superior a t_{cap} , es decir:

$$T_{cap} > 200ns \quad (3.6)$$

Finalmente, si tenemos en cuenta que los datos de salida son válidos solamente mientras RD_L este activo, el tiempo de lectura T_{read} está dado por:

$$T_{read} > T_{cap} \quad (3.7)$$

3.3 Etapa DSP

La etapa DSP se constituye en el núcleo de todo el sistema ya que realiza las siguientes funciones:

- Interpretación de las directivas de control realizadas en la interface de usuario.
- Control del factor de amplificación y desplazamiento vertical de la etapa de adaptación.
- Control de la etapa ADC.
- Procesamiento de la señal digital recibida por la etapa ADC.
- Impresión final en formato VGA.

Para la realización de estas tareas la etapa DSP se divide a nivel lógico en los siguientes bloques:

- **CONTROLADOR_ADC:** Genera la señal de control RD_L a partir de la señal INTR_L de la etapa ADC. Además muestrea la señal digital recibida de la etapa ADC y la mantiene fija para el correcto procesamiento del bloque NIOS_SYSTEM.

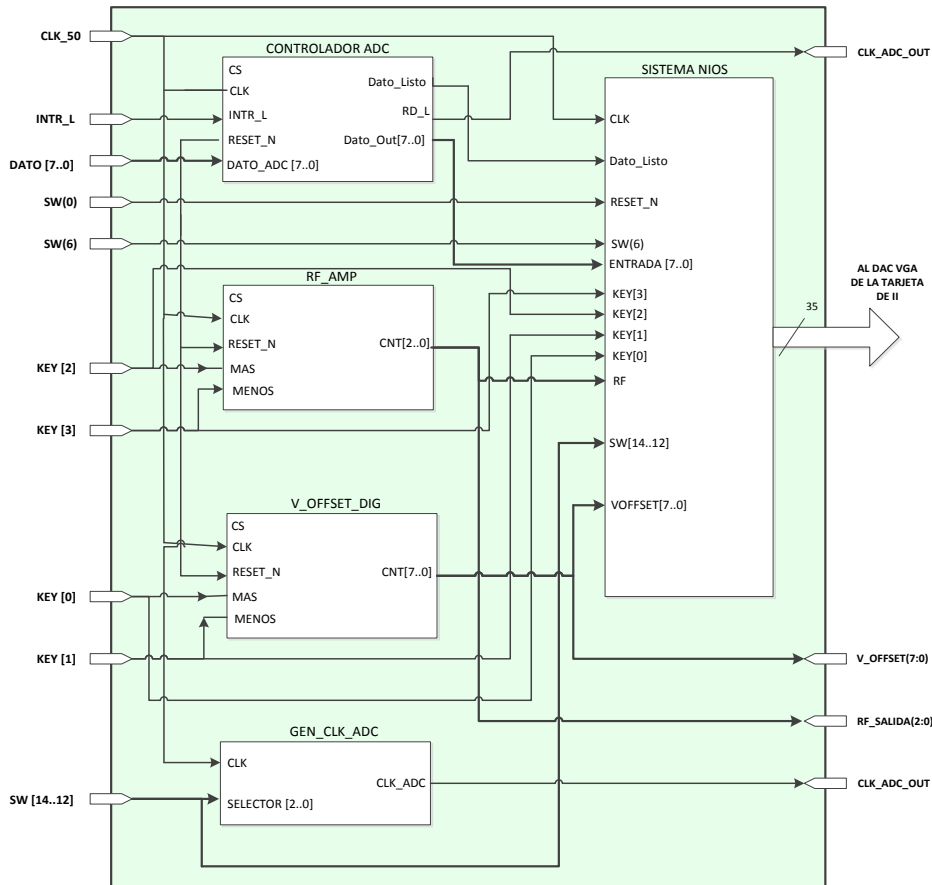


Fig. 3. 11 Diagrama de Bloques de la etapa DSP.

- **GEN_CLK_ADC:** Genera la señal de reloj de la etapa ADC. Selecciona la tasa del reloj en función de la señal digital SW(14:12) recibida desde la interface de usuario.

- **RF_AMP:** Genera la señal de control RF_AMP que maneja el factor de amplificación de la etapa de adaptación de señal. El valor de RF_AMP depende de las señales KEY(0) y KEY(1) manejadas desde la interface del usuario.
- **V_OFFSET_DIG:** Genera la señal de control V_OFFSET que maneja el desplazamiento vertical de voltaje en la etapa de adaptación de señal. El valor de V_OFFSET depende de las señales KEY(2) y KEY(3) manejadas desde la interface del usuario.
- **NIOS_SYSTEM:** Genera la señal de salida en formato VGA.

3.3.1 CONTROLADOR_ADC

Como su nombre lo indica, el objetivo de este bloque es controlar la etapa DSP, para esto se encarga de generar la señal de control RD_L a partir de la señal INTR_L. A nivel lógico el bloque es una maquina secuencial de seis estados cuyo comportamiento está determinado por el diagrama de la Fig.3.12.

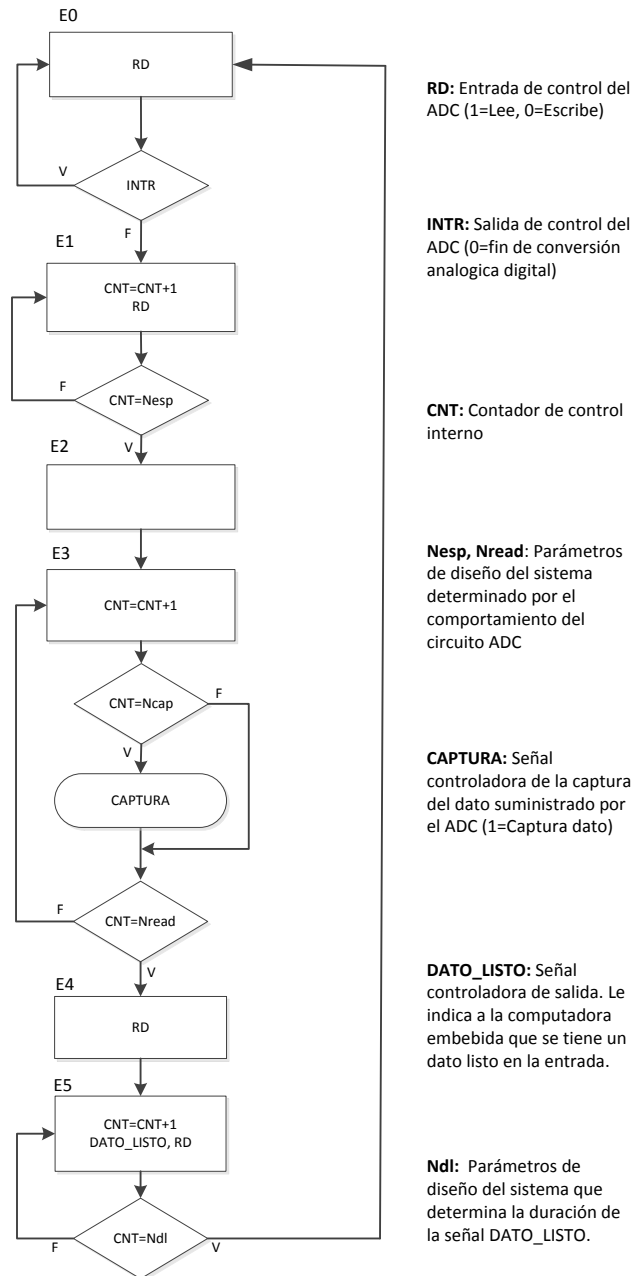


Fig. 3.12 Diagrama ASM del Controlador ADC

En el estado inicial E_0 el bloque está a la espera de la activación de la señal INTR_L que indica que el ADC tiene un dato listo, una vez detectado este evento el bloque pasa al estado de E_1 . El objetivo de E_1 es esperar un tiempo T_{esp} antes de activar la señal RD_L y capturar el dato, para esto la maquina secuencial espera N_{esp} flancos de reloj, es decir:

$$T_{esp} = N_{esp} * \left(\frac{1}{CLK} \right) \quad (3.8)$$

Una vez transcurrido el tiempo T_{esp} la máquina pasa al estado E_2 en donde la señal RD_L se activa con el objetivo de capturar el dato a la salida del ADC. No obstante, la captura no se la realiza en E_2 , sino en E_3 ya que es necesario esperar un tiempo prudencial t_{cap} para que el dato a la salida del ADC sea válido. Para esto la máquina secuencial espera N_{cap} flancos de reloj para habilitar la señal interna "captura" que muestrea la señal DATO_ADC y la carga en la salida DATO_OUT. En base de esto el tiempo de espera T_{cap} está dado por:

$$T_{cap} = N_{cap} * \left(\frac{1}{CLK} \right) \quad (3.9)$$

Una vez capturado el dato en E_3 es posible seguir con el flujo mediante la desactivación de la señal RD_L en el estado E_4 , cuando se cumplan los N_{read} flancos de reloj. En base de esto tenemos que:

$$T_{read} = N_{read} * \left(\frac{1}{CLK} \right). \quad (3.10)$$

$$N_{read} \geq N_{cap} \quad (3.11)$$

Una vez desactivada la señal RD_L en E_4 el bloque ADC comienza nuevamente el muestreo y conversión del próximo dato por lo cual aparentemente el proceso terminaría; sin embargo, la máquina secuencial tiene un estado adicional E_5 . El único objetivo de este estado es activar la señal de salida $dato_listo$ para que el bloque $NIOS_SYSTEM$ conozca cuando hay un dato nuevo a la salida del controlador ADC. La duración de este estado está determinada por el parámetro Ndl y debe ser un valor suficientemente alto para que el $NIOS_SYSTEM$ lo detecte.

CALCULO DE LA TASA DE MUESTREO

Analizando el esquema diseñado, el tiempo de muestreo está dado por el tiempo de conversión digital más la duración de los estados E1, E2 y E3, es decir:

$$T_s = T_c + T_{esp} + T_{read} \quad (3.12)$$

Remplazando (3.3), (3.8), y (3.10):

$$\begin{aligned} T_s &= 66 \left(\frac{1}{CLK_{ADC}} \right) + N_{esp} \left(\frac{1}{CLK} \right) + N_{read} \left(\frac{1}{CLK} \right) \\ &= 66 \left(\frac{1}{CLK_{ADC}} \right) + (N_{esp} + N_{read}) \left(\frac{1}{CLK} \right) \quad (3.13) \end{aligned}$$

En base de esto, la frecuencia de muestreo está dada por:

$$F_s = CLK * \left(\frac{CLK_{ADC}}{66 * CLK + (N_{esp} + N_{read}) * CLK_{ADC}} \right) \quad (3.14)$$

DISEÑO DE PARAMETROS NESP, NCAP, NREAD.

Para la selección adecuada de los parámetros de diseño es muy importante conocer el comportamiento en el tiempo del esquema utilizado en la etapa ADC.

Empecemos analizando el tiempo de espera T_{esp} y su contraparte digital N_{esp} que de acuerdo a lo expuesto están determinados por las ecuaciones (3.5) y (3.8). Si reemplazamos (3.8) en (3.5) tenemos que:

$$T_{esp} > 8 \left(\frac{1}{CLK_{ADC}} \right)$$

$$N_{esp} * \left(\frac{1}{CLK} \right) > 8 \left(\frac{1}{CLK_{ADC}} \right)$$

$$N_{esp} > 8 \left(\frac{CLK}{CLK_{ADC}} \right) \quad (3.15)$$

En lo referente al tiempo de captura T_{cap} y su equivalente digital N_{cap} su comportamiento está determinado por las ecuaciones (3.6) y (3.9). Si reemplazamos (3.9) en (3.6) tenemos:

$$T_{cap} > 200ns$$

$$N_{cap} * \left(\frac{1}{CLK} \right) > 200ns$$

$$N_{cap} > 200ns * CLK$$

Ahora, si tomamos en cuenta que el reloj utilizado para el controlador corresponde a una señal interna del FPGA de 50MHz, se tiene que:

$$N_{cap} > 10 \quad (3.16)$$

Finalmente, en lo referente a Nread, la única restricción que se tiene es que sea mayor a Ncap (Ecuaciones (3.11) y (3.7))

$$N_{read} \geq N_{cap} > 10 \quad (3.17)$$

3.3.2 GEN_CLK_ADC

La función del bloque GEN_CLK_ADC consiste en generar la señal de reloj de la etapa ADC. A nivel lógico el bloque es una máquina secuencial de dos estados cuyo comportamiento está determinado por el diagrama de la Fig. 3.13.

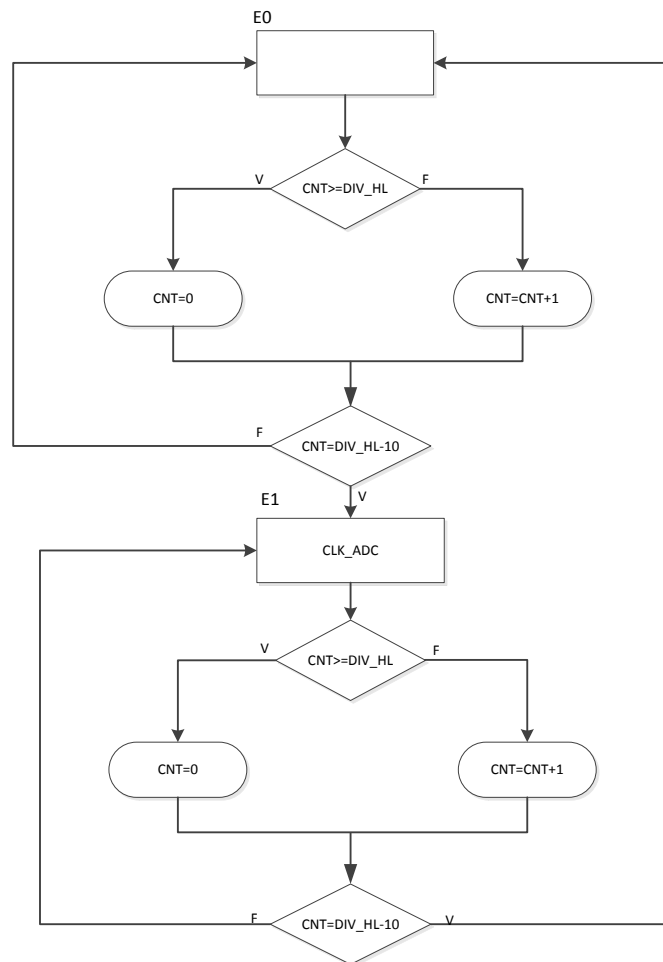


Fig. 3. 13 Diagrama ASM del Generador del reloj

La salida de reloj CLK_ADC presenta un valor de 0 y 1 en los estados E_0 y E_1 respectivamente, debido a que

ambos estados tienen una duración de DIV_HL flancos de reloj se tiene que:

$$CLK_{ADC} = \frac{CLK}{2 * DIV_HL} \quad (3.18)$$

Ahora, si la señal DIV_HL no es un valor constante sino una variable, es posible controlar dinámicamente la frecuencia de reloj de la señal de salida. Con este objetivo, el bloque también incluye un multiplexor que varía la señal DIV_HL (y por ende la frecuencia) en función de la señal de entrada $SELECTOR$ de 3 bits. Es importante acotar que la señal de reloj del ADC maneja de manera indirecta la frecuencia de muestreo, por lo cual con este mecanismo el usuario tiene la capacidad de manejar diferentes opciones para la resolución de la señal del osciloscopio en el dominio del tiempo. Finalmente, si el sistema utiliza un reloj de 50Mhz tenemos los valores mostrados en la Tabla VI.

Tabla VI *Valores de frecuencia de muestreo válidos para $N_{esp}=1000$, $N_{red}=25$

SELECTOR			DIV_HL	CLK_ADC	Fs
S14	S13	S12			
0	0	0	2500	10,0000	0,1510
0	0	1	500	50,0000	0,7460
0	1	0	250	100,0000	1,4695
0	1	1	125	200,0000	2,8531
1	0	0	83	301,2048	4,1733
1	0	1	62	403,2258	5,4295
1	1	0	50	500,0000	6,5574
1	1	1	39	641,0256	8,0998

3.3.3 RF_AMP:

En este bloque se genera la señal de control RF_AMP que maneja el factor de amplificación de la etapa de adaptación de señal. El valor de RF_AMP depende de las señales KEY(0) y KEY(1) manejadas desde la interface del usuario.

A nivel lógico el bloque es una máquina secuencial de cinco estados cuyo comportamiento está determinado

por el diagrama de la Fig. 3.14. En el estado inicial E_0 la máquina está a la espera de que el usuario pulse el botón + o – para según aquello pasar al estado E_1 o E_2 respectivamente. Además, en este estado la salida es inicializada al valor por defecto cnt_0 . En el estado de suma E_1 la salida es aumentada en una unidad, siempre y cuando no se haya alcanzado el valor máximo ya que de ser así el valor debe mantenerse para evitar un sobreflujo. De manera análoga en el estado de resta E_2 la salida es disminuida en una unidad, siempre y cuando no sea cero ya que la misma representa un número entero. Tanto en E_1 como en E_2 el siguiente estado es E_3 , donde la máquina va a esperar “ESPERA” flancos de reloj antes de pasar al estado E_4 . El estado E_3 es necesario ya que las señales + y – son manipuladas por el usuario y la frecuencia de muestreo es demasiado rápida en comparación a las reacciones humanas. Finalmente, el estado E_4 es similar a E_0 , la diferencia radica en que en E_4 la salida se mantiene y no es cargada con el valor inicial cnt_0 .

En definitiva el bloque es un sumador/restador que genera la señal selectora Rf_amp , en base de eso el parámetro MAX determinara el número máximo de opciones de amplificación del sistema.

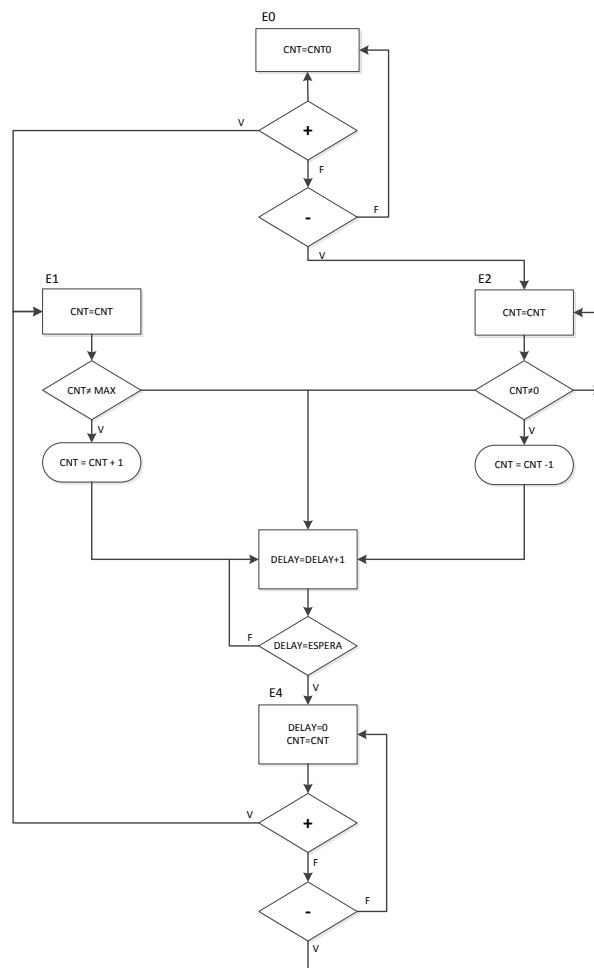


Fig. 3. 14 Diagrama ASM del Bloque de Amplificación

3.3.4 V_OFFSET_DIG:

Este bloque genera la señal de control V_OFFSET que maneja el desplazamiento vertical de voltaje en la etapa de adaptación de señal. El valor de V_OFFSET depende de las señales KEY(2) y KEY(3) manejadas desde la interface del usuario. A nivel lógico el comportamiento del bloque es idéntico a RF_AMP, la diferencia radica en los valores de sus parámetros MAX y cnt₀. En este caso MAX definirá la granularidad de paso ΔV_OFFSET de acuerdo a:

$$\Delta V_{offset} = \frac{5}{2^{MAX}} \quad (3.19)$$

Por otro lado, el parámetro cnt₀ define el valor inicial del offset. Debido a que el ADC procesa señales entre 0 y 5V en condiciones ideales el valor inicial del offset es 2.5V. No obstante, hay que tener en cuenta que el circuito analógico puede generar un valor de offset ya que los elementos reales no se comportan de manera ideal. En base de esto es conveniente definir la variable OFFSET_OPAMP con la finalidad de corregir este error

una vez calculado en la etapa de pruebas (Capitulo 5).

Así tendríamos que:

$$CNT_0 = 128 - OFFSET_OPAMP \quad (3.20)$$

3.3.5 NIOS_SYSTEM

El objetivo principal del NIOS_SYSTEM generar la señal de salida del osciloscopio en formato VGA. El bloque básicamente se trata de una computadora embebida en un bloque HDL mediante el uso del sistema NIOS proveído por Altera. Por esta razón el funcionamiento específico del bloque en realidad depende del software de alto nivel a implementarse en dicha computadora. En base de esto nuestro punto de partida del diseño del software es que abarca las siguientes tareas:

- Impresión de texto descriptivo y cuadrícula en pantalla.
- Formación de los símbolos a partir de las señales digitales muestreadas por el ADC.

- Impresión de símbolos en pantalla mediante dos modos de operación:
- Modo de Muestreo Continuo
- Modo de Fotografía

Para la realización de las tareas antes mencionadas el software diseñado realiza las funciones especificadas por el diagrama de bloques de la Fig. 3.15 . A continuación se especificaran las características de cada uno de estos bloques.

Borrado de pantalla

Como su nombre lo indica el objetivo de este módulo es borrar el contenido gráfico en Pantalla. En nuestro caso, tanto la impresión de la cuadrícula como la de los símbolos es realizada mediante el registro de pixeles de la computadora NIOS, por lo cual el borrado de pantalla básicamente consiste en borrar este registro. Para esto el modulo utiliza la función `alt_up_pixel_buffer_dma_clear_screen` de la librería `altera_up_avalon_video_pixel_buffer_dma.h`.

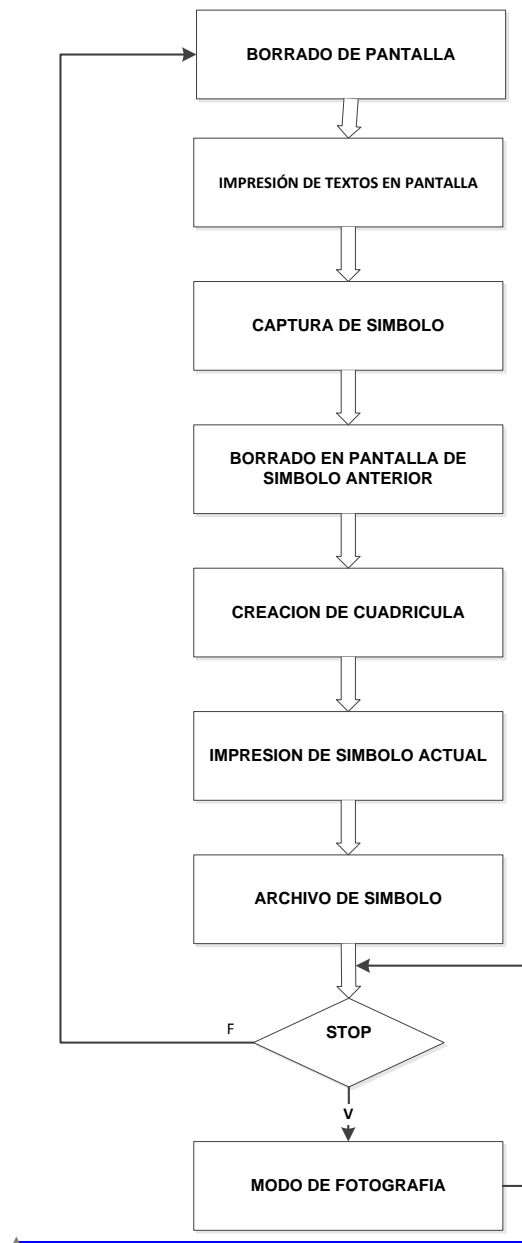


Fig. 3. 15 Diagrama de Flujo del desarrollo de software

Código de campo cambiado

Impresión de Textos en pantalla

El objetivo de esta fase del software es graficar los textos de presentación del Osciloscopio en pantalla, para esto se utiliza el registro de caracteres VGA del sistema NIOS. Entre los detalles más importantes a mostrar en pantalla se tienen:

- Tipo de amplificación (depende de la salida del bloque RF_AMP).
- Voltaje Offset (depende de la salida del bloque V_OFFSET_DIG).
- Escalamiento vertical (1 div=1 voltio).
- Escalamiento horizontal (depende del selector de la frecuencia de muestreo, es decir S14, S13 y S12).
- Modo de Operación (Muestreo Continuo).

Captura de Símbolo

El objetivo de este bloque es la formación del símbolo a imprimir en función de los datos muestreados en la etapa

ADC, lo cual se logra mediante la implementación del algoritmo de la Fig. 3.16.

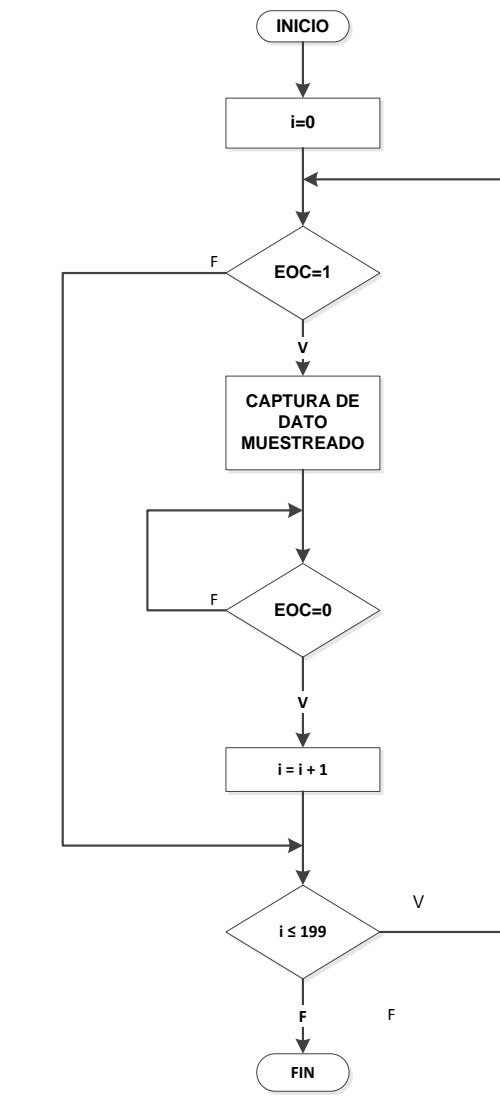


Fig. 3. 16 Diagrama de Flujo de la captura del Símbolo

Código de campo cambiado

En el esquema mostrado el sistema espera hasta que el bloque ADC indique que tiene un dato listo mediante la activación de la señal EOC. Cuando esto sucede el dato es capturado y escalado para su posterior impresión en pantalla, el valor resultante de este proceso se almacena en la variable dato que ira llenando de manera progresiva el arreglo V que representa al símbolo formado. Al final del proceso se tiene que la variable V contiene la información del símbolo formado por 200 muestreos, este se debe a que la resolución vertical diseñada corresponde a 200 pixeles.

Borrado en pantalla del símbolo anterior

Antes de imprimir el símbolo formado es necesario borrar el símbolo anterior que aún se encuentra en pantalla, esto se logra mediante la implementación del algoritmo de la Fig. 3.17.

La lógica de este bloque es básicamente la misma que se utiliza en la impresión del símbolo; de hecho, en realidad se trata de imprimir nuevamente el símbolo

(Que ahora debe estar cargado en la variable V_ant)
pero con el mismo color del fondo de pantalla.

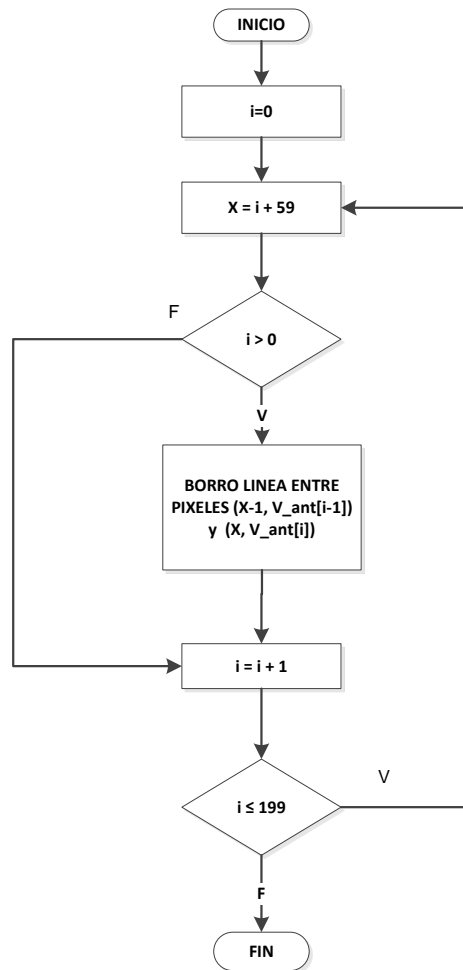


Fig. 3. 17 Diagrama de Borrado del símbolo anterior

Para la impresión del símbolo se utiliza la función `alt_up_pixel_buffer_dma_draw_line` de la librería `altera_up_avalon_video_pixel_buffer_dma.h` que permite graficar una línea mediante el registro de píxeles del NIOS II. En sí, la idea del algoritmo es recorrer los diferentes píxeles ($x, V_ant[i]$) y unirlos con líneas formando de esta manera la gráfica de la señal deseada.

Creación de la Cuadrícula

El objetivo de este bloque es formar la cuadrícula que sirve como guía para la visualización de las señales en el osciloscopio, esto se logra mediante la implementación del diagrama de la Fig. 3.18. En este módulo se imprimen progresivamente las líneas horizontales y verticales que forman la cuadrícula. Como se puede inferir del diagrama la distancia entre línea y línea es de 20 píxeles, razón por la cual cada división tendrá dicha cantidad de píxeles.

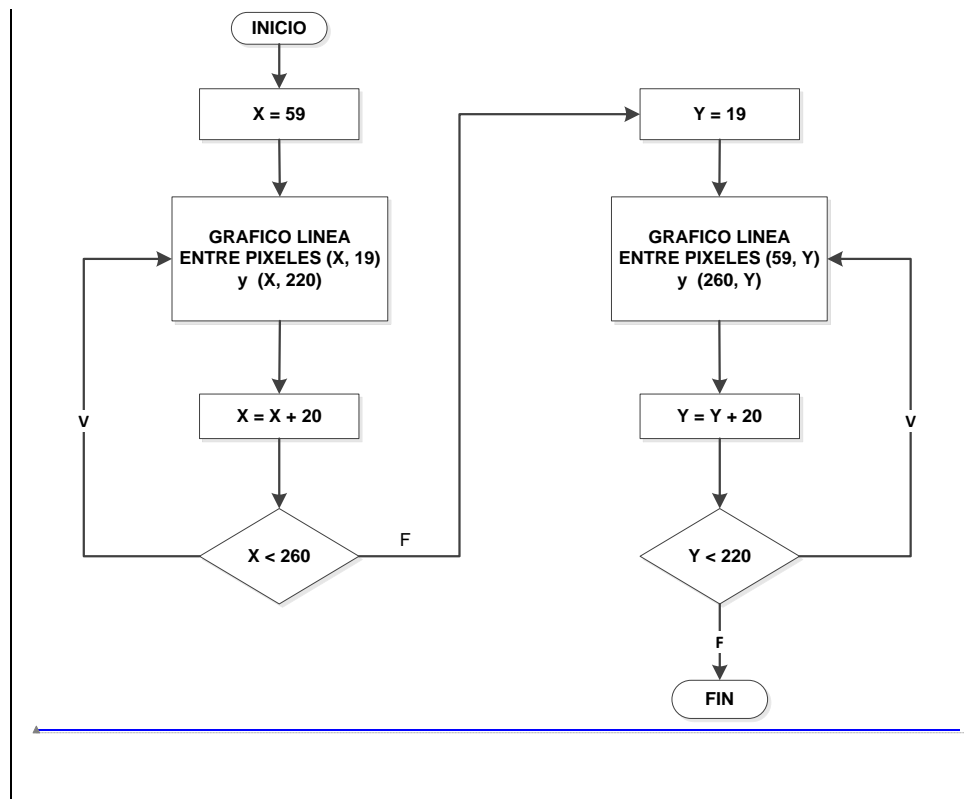


Fig. 3. 18 Creación de Cuadrícula

Impresión de Símbolo Actual

El funcionamiento de este módulo es básicamente el mismo que el usado en el proceso de borrado del símbolo anterior. De hecho, solamente existen dos diferencias:

- En este caso el arreglo a graficar es V y no V_ant.

Código de campo cambiado

- El color de las líneas a graficar debe ser diferente al fondo de pantalla y a la cuadrícula.

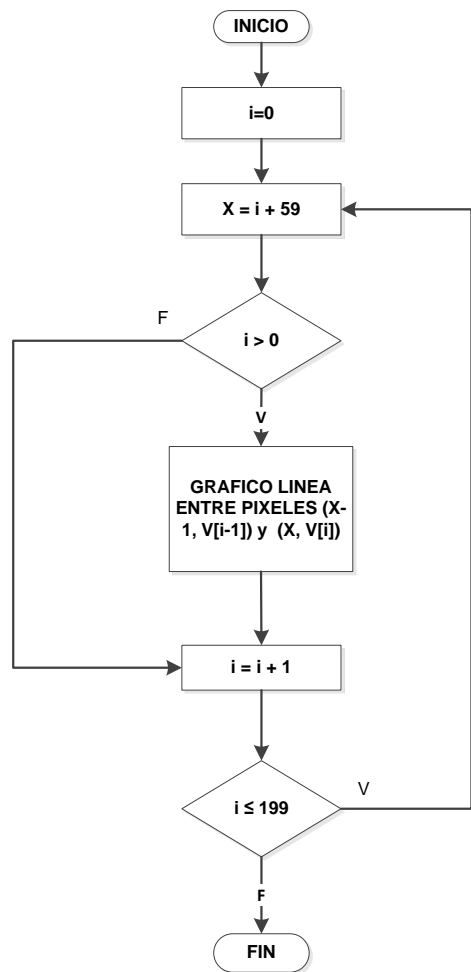


Fig. 3. 19 Diagrama para impresión del símbolo actual

Código de campo cambiado

Archivo de Símbolo

Una vez impreso el símbolo actual es necesario archivarlo para cuando sea necesario borrarlo en pantalla en el proceso de carga del siguiente símbolo. Este se logra cargando el arreglo V en V_ant ($V_{ant} = V$).

Modo de Fotografía

Cabe recordar que una de las especificaciones originales del diseño es poseer dos modos: muestreo continuo y fotografía. El objetivo del modo de fotografía es dejar congelada la imagen en pantalla para que el usuario pueda realizar los análisis pertinentes. Desde este punto de vista en este modo no debería hacerse acción alguna, sin embargo hay que tener en cuenta en la fase de impresión de texto se definió que el modo era el de muestreo continuo. En definitiva, es esta etapa el software sobre escribe en pantalla la información del modo indicando que ahora está en modalidad fotografía.

El software solo entra en esta etapa si se ha culminado el archivo del símbolo y la entrada STOP permanece activada (SW(6)).

CAPITULO 4

4. Implementación del Sistema

En este capítulo se detallara la implementación de hardware y software del diseño expuesto en el capítulo anterior, básicamente se expondrán los parámetros utilizados en la implementación.

4.1 Implementación de Hardware

Tal como se expuso en el anterior capitulo el sistema en general tiene 3 etapas: adaptación de la señal, muestreo y conversión analógica/Digital (ADC) y procesamiento digital de señales (DSP, por sus siglas en ingles). A nivel de hardware, las dos primeras etapas fueron implementadas en un circuito electrónico independiente, mientras que para la tercera etapa y la interface de usuario se utilizó la tarjeta DE2 de Altera. De manera más específica, la etapa DSP se implementa en el FPGA de la tarjeta DE2, mientras que para la interface de usuario se utilizan los pulsadores KEY(0), KEY(1), KEY(2) y

KEY(3) y los interruptores SW(14), SW(13) y SW(12) de dicha tarjeta, como se explicó en el capítulo 2.



Fig. 4.1 Implementación de circuito analógico

El diagrama físico de la tarjeta electrónica implementada se encuentra en la Fig. 4.1, mientras que los valores específicos de los elementos utilizados están detallados en la tabla VII. Es de notar que en el caso de las resistencias los valores difieren a los teóricos debido a la disponibilidad en el mercado. Este es un punto importante a tener en cuenta ya que se esperaría que los factores de amplificación también se vieran afectados por estas divergencias.

Tabla VII Valores de elementos utilizados

ETAPA	ELEMENTO	VALOR	TOLERANCIA
DESPLAZAMIENTO VERTICAL	R1	20K	±5%
	R2	33K	±5%
	R3	10K	±5%
	R4	20K	±5%
AMPLIFICACION	R5	10K	±5%
	R6	20K	±5%
	R11	1000K	±5%
	R12	470K	±5%
	R13	150K	±5%
	R14	100K	±5%
	R15	47K	±5%
	R16	30K	±5%
	R17	20K	±5%
R18	10K	±5%	
ADAPTACION FINAL	R7	20K	±5%
	R8	20K	±5%
	R9	30K	±5%
	R10	10K	±5%

4.2 Implementación de Software / Hardware

En la etapa DSP de nuestro proyecto existen dos tipos de desarrollo de Software:

- Hardware programado en alto nivel mediante un lenguaje de descripción de hardware (HDL).

- Software programado en lenguaje de alto nivel, específicamente en C++.

Para la implementación del hardware de alto nivel se utilizaron las herramientas de Quartus II de altera. Con el uso de este programa se implementó en VHDL el diseño especificado de los bloques: CONTROLADOR_ADC, GEN_CLK_ADC, RF_AM y V_OFFSET_DIG. Por otra parte, para la implementación de la computadora del bloque NIOS SYSTEM se utilizó el sistema de computadora media DE2, proveído por el Quartus II. Los parámetros de diseño utilizados en la implementación a nivel HDL son los siguientes están especificados en la Tabla VIII y cumplen las condiciones de diseño expuestas en el capítulo 3. Cabe acotar que para la definición del parámetro CNT_0 fue necesario esperar a la etapa de pruebas ya que depende del valor experimental OFFSET_OPAMP (Ver sección 5.1).

Por otro lado, para la implementación del software de alto nivel se utilizó el programa NIOS II IDE, en el cual se plasmó el diseño del bloque NIOS SYSTEM desarrollado en el capítulo 3. Un detalle completo del código programado en C++ se puede revisar en el Anexo A.

Tabla VIII Definición de parámetros utilizados en el software

BLOQUE	PARAMETRO	VALOR
GEN_CLK_ADC	DIV_HL	Valores dependientes de la señal selectora de acuerdo a la tabla VI
CONTROLADOR_ADC	N_ESP	1000
	N_CAP	20
	N_READ	25
	N_DL	200
V_OFFSET_DIG	CNT ₀	115
	MAX_contador	255
	DELAY	0
	MAX_DELAY	25000000
	ncontador	0
RF_AMP	contador	0
	MAX_contador	7
	DELAY	0
	MAX_DELAY	25000000

CAPITULO 5

5. Pruebas y Resultados

Para la definición de las pruebas del sistema implementado se han tomado en cuenta los siguientes objetivos:

- Calibrar el Sistema
- Definir los parámetros de operación del sistema
- Formular las conclusiones y recomendaciones finales del proyecto

En base de esto, las pruebas realizadas fueron las siguientes:

1. Medición y Corrección de Voltaje Offset del circuito Electrónico.
2. Variación del Voltaje Offset Digital.
3. Medición de factores de Amplificación.
4. Medición de frecuencia de muestreo generada.
5. Variación de la frecuencia de la señal de entrada.

En el transcurso del capítulo se expondrán en mayor detalle los resultados de las pruebas realizadas para finalmente definir los parámetros de operación del sistema en base a las mismas.

5.1 Medición y Corrección de Voltaje Offset del circuito Electrónico.

El objetivo principal de esta prueba es el cálculo y corrección del Offset del circuito electrónico en el sistema, es decir descubrir el valor de la variable OFFSET_AMP de la ecuación (3.20). Por esta razón esta fue la primera de las pruebas realizadas ya que en las siguientes el offset del circuito electrónico ya estaba corregido.

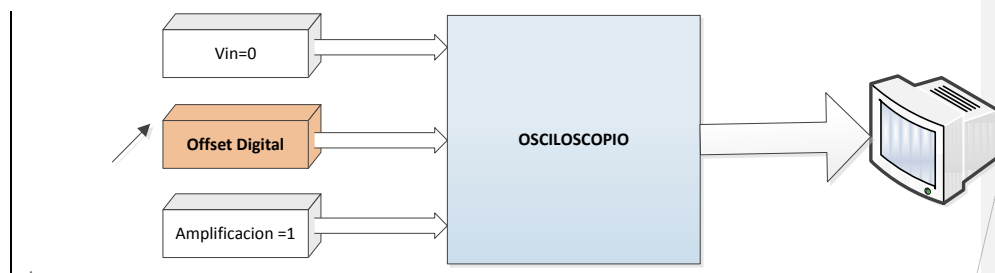


Fig. 5.1 Esquema de pruebas de voltaje offset

Para la ejecución de estas pruebas se definieron las siguientes condiciones:

1. El voltaje de entrada fue fijado en 0V.
2. El factor de amplificación utilizado fue de 1.
3. El software de alto nivel del osciloscopio fue alterado para que se muestre en pantalla el valor instantáneo de la señal de salida y el valor del voltaje offset digital insertado (ANEXO B).
4. El valor del offset fue variado manualmente mediante la manipulación de los pulsadores de la interface de usuario.

Si tomamos en cuenta los puntos 1 y 2 y las ecuaciones (3.1), (3.19) y (3.20) la salida del sistema en este caso correspondería al voltaje offset digital insertado más el voltaje offset del circuito electrónico (OFFSET_AMP). Así, gracias a la alteración del software ejecutada en el punto 3 y la variación del offset digital indicada en el punto 4 se puede calcular fácilmente el valor de OFFSET_AMP, la tarea se reduce a documentar los datos mostrados por el osciloscopio en pantalla. Los datos de estas pruebas se encuentran en la Tabla IX y de los mismos se puede inferir que el valor de OFFSET_AMP corresponde a -11. Finalmente, en base de esta prueba el parámetro CNT_0 del bloque V_OFFSET_DIG en la etapa DSP debe ser definido en 115, dicho valor también tiene que ser tomado en cuenta en el software final si

se desea mostrar el valor del offset insertado digitalmente por el usuario.

Tabla IX Pruebas de Calibración Offset

VOLTAJE OFFSET (Salida del bloque V_OFFSET_DIG)	VOLTAJE MEDIDO
127	-0,529412
126	-0,45098
125	-0,411765
124	-0,372549
123	-0,333333
122	-0,294117
121	-0,254902
120	-0,215686
119	-0,176471
118	-0,137255
117	-0,058236
116	-0,019608
115	0,019608

5.2 Variación del voltaje offset digital

La finalidad de esta prueba fue probar el comportamiento del sistema ante la variación del offset, para lo cual se documentó el

paso del offset observado y se calculó el error del mismo con respecto al valor teórico expresado en (3.19). Las condiciones para la ejecución de estas pruebas son las mismas que las de la sección anterior, la única diferencia es que en este caso el offset del circuito electrónico ya estaba corregido.

De las mediciones realizadas (ANEXO C) se observa que el valor medio de paso del offset es de 0,03921567 lo cual corresponde a un error del 0.39% con respecto al valor teórico. Para explicar este error debemos tener en cuenta que las resistencias utilizadas en la conversión digital-analógica no son ideales, así como los valores digitales 1 y 0 no son exactamente 5 y 0 Voltios respectivamente.

5.3 Medición de factores de Amplificación

La finalidad de esta prueba fue evaluar el comportamiento del sistema ante la variación de los factores de amplificación, para esto se midieron los valores experimentales de amplificación y se los comparo con los valores teóricos en base a las resistencias utilizadas Tabla VI. Para el cálculo experimental de los factores de amplificación se utilizaron señales sinusoidales y se compararon sus valores pico a pico a la entrada y salida del

sistema. Para la medición del voltaje pico a pico de la entrada se utilizó un osciloscopio digital de alta precisión, mientras que para la medición de la salida se realizó una inspección visual de lo mostrado en pantalla por nuestro desarrollo. Los resultados de estas pruebas se resumen en la Tabla X, mientras que un registro fotográfico de estas pruebas se encuentra en el ANEXO D.

Tabla X Valores medidos en la Amplificación

MODO	Rf medido [kΩ]	Ax TEORICO	Vin [V]	SENAL DEGRADADA	Vout [V]	Ax PRACTICO	Error
1	10	1	1	NO	1	1	0,00%
2	20	2	1	NO	2	2	0,00%
3	30	3	1	NO	2,77	2,77	7,67%
4	47	4,7	1	NO	3,8	3,8	19,15%
5	100	10	0,25	SI	-	-	-
6	150	15	0,25	SI	-	-	-
7	470	47	0,05	SI	-	-	-
8	1000	100	0,05	SI	-	-	-

Tabla con formato

De lo observado experimentalmente se tienen dos resultados inesperados:

- El porcentaje de error de los factores de amplificación con respecto al valor teórico es bastante elevado para valores de amplificación altos.
- La señal de entrada se degrada ostensiblemente con los 4 últimos factores de amplificación.

De lo analizado estos errores se deben principalmente a:

- Los valores de las resistencias no son exactos.
- El comportamiento del multiplexor CD 4067 utilizado para la selección de R_f no es ideal y mientras mayor es la resistencia escogida mayor el error.

En base de esto en la sección de parámetros del sistema solo se incluyen los valores correspondientes a los cuatro primeros factores.

5.4 Medición de frecuencia de muestreo generada

La finalidad de esta prueba es documentar las diferentes frecuencias de muestreo generadas por el sistema. Para esto se varió la señal selectora de frecuencia de la etapa DSP y se midió la frecuencia de la señal INTR_L generada por la etapa ADC. Para estas mediciones se utilizó el osciloscopio digital de alta precisión

Tektronix 1002B y sus resultados se resumen en la Tabla XI. En este caso el porcentaje de error es causado principalmente por el tiempo de conversión del CI ADC0804.

Tabla XI Tabla de Frecuencias Obtenidas por el sistema

s14	s13	s12	Frecuencia	Frecuencia Medida	Error
0	0	0	0,151045994	0,1387	8,17%
0	0	1	0,745990302	0,6931	7,09%
0	1	0	1,469507715	1,383	5,89%
0	1	1	2,853067047	2,755	3,44%
1	0	0	4,173274351	4,132	0,99%
1	0	1	5,42947117	4,975	8,37%
1	1	0	6,557377049	6,136	6,43%
1	1	1	8,099789405	7,812	3,55%

5.5 Variación de la frecuencia de la señal de entrada

El objetivo de esta prueba es validar el correcto funcionamiento del osciloscopio ante señales de diferentes rangos de frecuencia. Teniendo en cuenta este objetivo se definieron las siguientes condiciones:

1. El voltaje de offset fue definido al valor por defecto.
2. El factor de amplificación utilizado fue de 1.

3. Se definió una señal sinusoidal de 1 voltio pico a pico como entrada del sistema. La frecuencia de dicha señal fue variada en un rango entre 1Hz y 2KHz

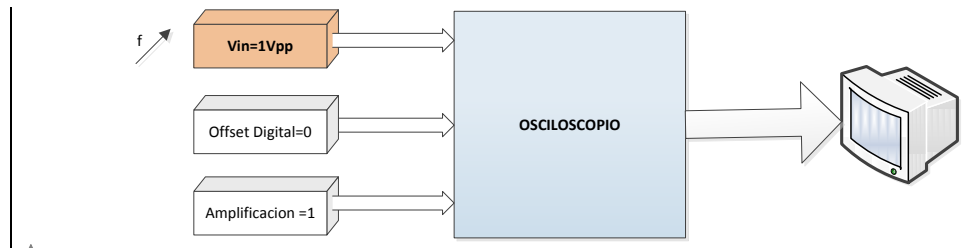


Fig. 5. 2 Pruebas de Variación de frecuencia de la señal de entrada

En definitiva el escenario consistió en variar la frecuencia de la señal de entrada y observar como el osciloscopio desarrollado la mostraba en pantalla en sus diferentes modos de frecuencia. Para mayor referencia una documentación fotográfica de las pruebas desarrolladas se encuentra en el ANEXO F.

Si bien es cierto el teorema de Nyquist indica que basta con que la frecuencia de muestreo sea el doble de la señal muestreada[13], la experiencia gráfica en dicho escenario es muy pobre. De hecho, en base a las pruebas realizadas nos parece recomendable que la

Código de campo cambiado

frecuencia de muestreo sea al menos 8 veces mayor que la frecuencia de la señal de entrada Fig. 5.3.

Por otro lado, si bien es cierto a mayor frecuencia de muestreo existe un mayor detalle de la señal, es necesario considerar que el osciloscopio tiene una resolución horizontal de 200 píxeles, por lo cual si la frecuencia de muestreo es mucho mayor solo se visualiza una pequeña parte de la señal. Por ejemplo, si se quiere visualizar una señal de 1 Hz con una frecuencia de muestro de 8kHz lo que se visualizará es algo muy cercano a una línea recta. En base de esta observación consideramos recomendable que al menos 1 periodo completo de la señal sea visualizado en la pantalla del osciloscopio, esto se logra siempre y cuando la señal de muestreo sea como máximo 200 veces mayor que la señal de entrada Fig. 5.4.

Teniendo en cuenta las recomendaciones anteriormente indicadas y los diferentes valores de frecuencia de muestro, la frecuencia de la señal de entrada debe oscilar entre 0.75 Hz y 976Hz.

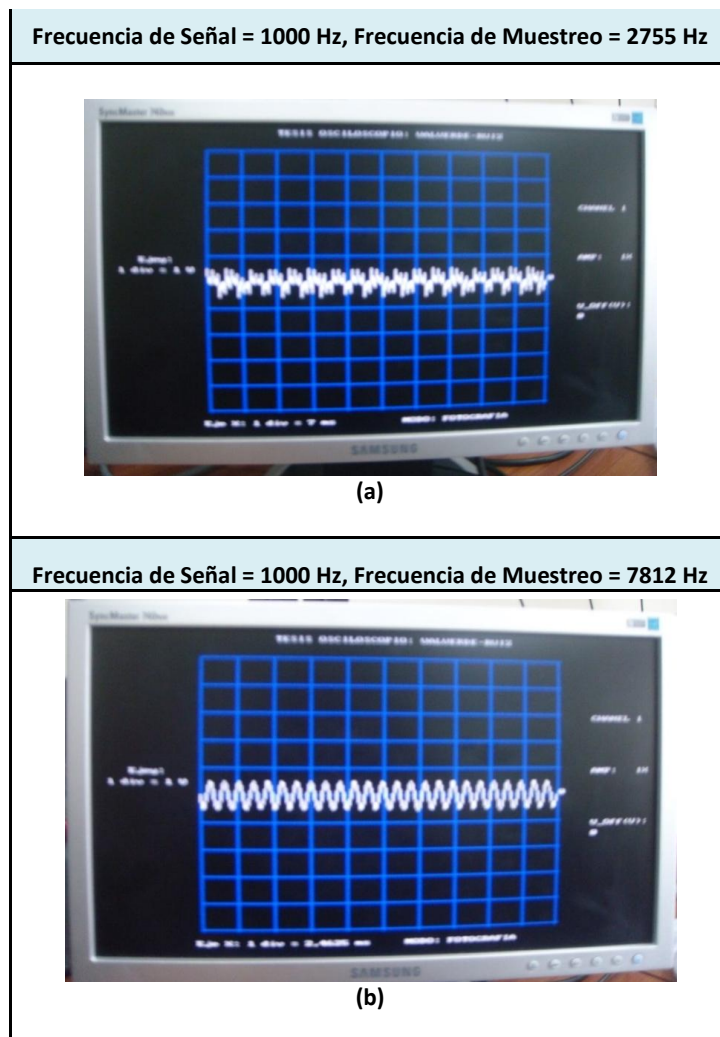


Fig. 5.3 Graficas de Salida: **(a)** Frecuencia de muestreo 2.7 veces la frecuencia de la señal. **(b)** Frecuencia de muestreo 7.8 veces la frecuencia de la señal.

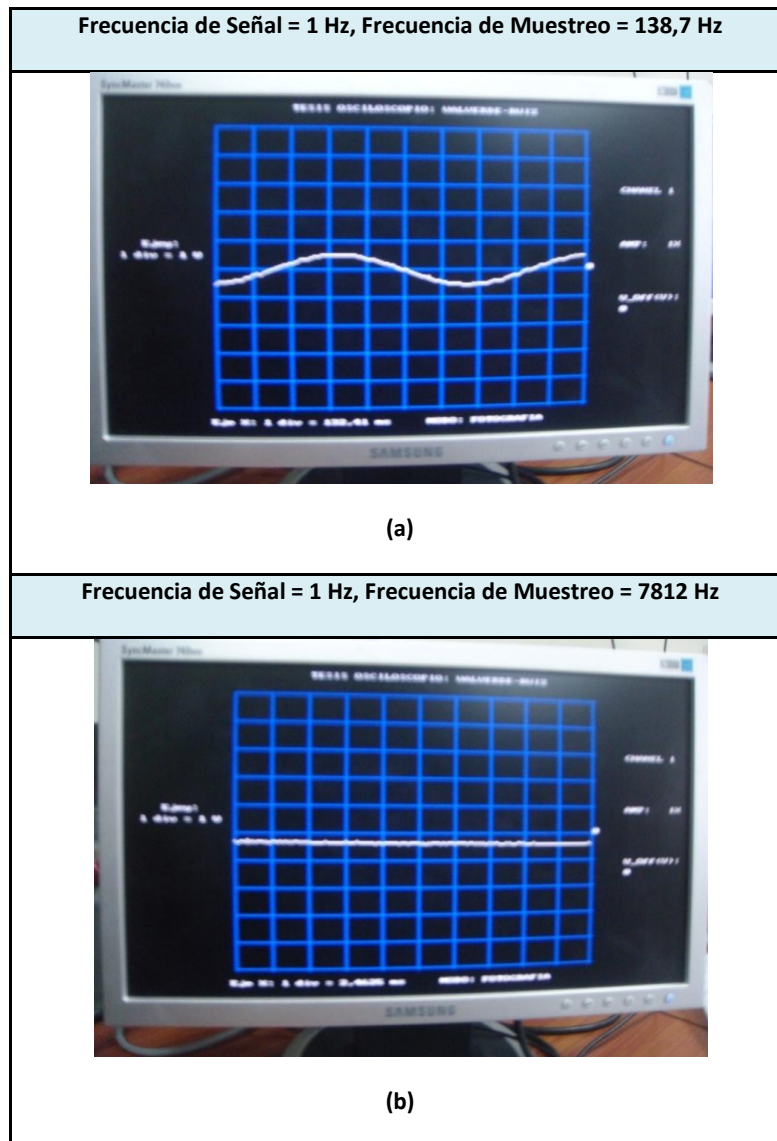


Fig. 5. 4 Graficas de Salida: **(a)** Frecuencia de muestreo 138 veces la frecuencia de la señal. **(b)** Frecuencia de muestreo 7800 veces la frecuencia de la señal.

5.6 Parámetros de operación del sistema

En base a la experiencia de las diferentes pruebas detalladas en este capítulo los parámetros de operación del sistema resultante están determinados por:

Tabla XII Parámetros finales del sistema

SEÑAL DE ENTRADA			
Parámetro	Mínimo	Máximo	
Voltaje	-5	5	
Frecuencia	0,75 Hz	976HZ	
MODOS DE OPERACION			
SW6	Modo	Descripción	
0	Muestreo Continuo	La señal se visualiza de manera continua en la pantalla del osciloscopio	
1	Fotografía	La última muestra de la señal se queda fija en la pantalla	
FRECUENCIA DE MUESTREO			
s14	s13	s12	Frecuencia(khz)
0	0	0	0,1387
0	0	1	0,6931

0	1	0	1,383
0	1	1	2,755
1	0	0	4,132
1	0	1	4,975
1	1	0	6,136
1	1	1	7,812
FACTORES DE AMPLIFICACION			
MODO		FACTOR	
1 (Por defecto)		1x	
2		2x	
3		2,77x	
4		3,8x	

CONCLUSIONES

1. Durante el presente proyecto de tesis se desarrolló de manera satisfactoria un osciloscopio mediante la utilización de la tarjeta DE2 de Altera, la cual contiene un procesador NIOS II. Los parámetros de operación de dicho osciloscopio están determinados por la tabla XI.
2. La fase de adaptación de señal genera un voltaje offset extra en la entrada digital. Esto se debe a que los elementos electrónicos utilizados no son ideales por lo cual concluimos que siempre existirá un error remanente y su magnitud es proporcional a la precisión de la etapa de adaptación de señal.
3. El valor medio del paso del offset digital es de 0,03921567, lo cual corresponde a un error del 0.39% con respecto al valor teórico. Esto se debe a que las resistencias utilizadas en la conversión digital-analógica no son ideales, así como los valores digitales 1 y 0 no son exactamente 5 y 0 Voltios respectivamente.
4. En base a las pruebas de amplificación realizadas se observa que el error de la amplificación con respecto al valor teórico es mayor conforme aumenta la magnitud de amplificación, hasta llegar a un punto que inclusive la señal se degrada. Concluimos que este

comportamiento se debe principalmente al multiplexor CD 4067 utilizado para la selección de R_f , el cual difiere en mucho con un multiplexor ideal sobre todo para valores de resistencia elevados.

5. De acuerdo a las pruebas de frecuencia de muestro del osciloscopio este opera en 5 modos que van desde los 138Hz y los 7.8kHz, mientras que el error con respecto al valor teórico oscila entre el 0.63% y 8.1%. Concluimos que tanto los valores bajos de frecuencia de muestreo como el porcentaje de error se debe al tiempo de conversión del CI ADC0804 utilizado en la etapa de conversión analógica-digital.
6. Si bien es cierto el teorema de Nyquist indica que basta con que la frecuencia de muestreo sea el doble de la señal muestreada, para una experiencia grafica plena la frecuencia de muestreo del osciloscopio debe ser al menos 8 veces mayor a la frecuencia de la señal de entrada.
7. Si bien es cierto a mayor frecuencia de muestreo existe un mayor detalle de la señal muestreada es necesario tener en cuenta la resolución horizontal H del osciloscopio. En base de esto, para una visualización grafica plena la frecuencia de muestreo no debe superar más de H veces la frecuencia de la señal de entrada.

RECOMENDACIONES

1. Debido a que fase de adaptación de señal genera un voltaje offset extra en la entrada digital creemos recomendable que en trabajos similares a los nuestros (Siempre que se tenga un procesamiento electrónico previo a una etapa DSP) se incluya una etapa de medición y calibración del voltaje offset remanente como la desarrollada en la sección 5.1. Sobre este punto también consideramos recomendable que la corrección de dicho offset no sea solamente realizada en la etapa DSP como es nuestro caso, sino que se divida en una corrección gruesa en el circuito electrónico y una corrección fina a nivel DSP.
2. Debido a los diferentes errores insertados por la inexactitud de los valores de la resistencia consideramos recomendable que para futuros trabajos se preste especial atención en los valores de tolerancia de las mismas, sobre todo si se desea realizar un desarrollo más profundo y elaborado en base a nuestro proyecto académico. Sobre este punto también es recomendable tener en cuenta que en el mercado local no se tiene mucha disponibilidad de elementos con baja tolerancia.

3. Debido al bajo rendimiento del circuito de amplificación utilizado mediante el multiplexor CD 4067, consideramos recomendable que para futuros desarrollos similares se implemente una opción alterna a esta.

4. Si bien es cierto nuestro proyecto es netamente académico y los valores de frecuencia de muestreo satisfacen nuestras expectativas, en aplicaciones reales y más elaboradas estas limitantes deben tomarse en cuenta. En base de esto consideramos recomendable utilizar un circuito integrado de mejor rendimiento que el ADC0804. Sobre este punto sería recomendable considerar dos factores: la baja disponibilidad de CIs ADCs en el mercado local y que el controlador depende directamente del chip utilizado, por lo cual sería necesario desarrollar un nuevo diseño al respecto.

ANEXOS

ANEXO A

```
#include <stdlib.h>
#include <stdio.h>
#include <io.h>
#include <system.h>
#include <math.h>
#include <altera_up_avalon_video_pixel_buffer_dma.h>

void write_pixel(int x, int y, int colour);
void clear_screen(alt_up_pixel_buffer_dma_dev *pixel_buffer);
void crear_cuadrícula(alt_up_pixel_buffer_dma_dev *pixel_buffer);
void borra_captura ();
int captura_dato();
int captura_offset();
int captura_amp();
void mostrar(int V[], int desplazamiento);

volatile char * character_buffer = (char *) 0x01080000; // direccion
de Buffer de Caracteres
void VGA_text(int x, int y, char * text_ptr);// funcion para
escribir en el vga
int V[280],V_ant[280],i,j,desplazamiento, dato, a, x;

//VARIABLES PARA ESCRIBIR EN EL VGA
short pixel_color;
int offset, row, col;
int x1=54*4,s1=13*4,x2=70*4,y2=17*4;
short pixel_color;
int x=70,y=15;
char * text_ptr;
char V_offset_char[40];
char amp_char[40];
char ejey_char[40];
```

```

char modo[60];
char text_top_row[40] = "CHANEL 1\0";
char voltaje[40]="V_Off(V) :";

void main()

{
    volatile int *pKEY = (int*)0x1083070;
    volatile int *pSWITCH= (int*)0x1083060;
    volatile int *pGPIO0= (int*)0x010830e0;
    volatile int *pEOC= (int*)0x010830f0;
    volatile int *pLEDR = (int*)0x01083020;
    volatile int *vga_addr;
    volatile int *poffset= (int*)0x00000000;
    volatile int *pRF= (int*)0x00000010;
    int sw,GPIO0,EOC,LEDR, datolis, stop, frec;
    alt_up_pixel_buffer_dma_dev *pixel_buf_dev;
        pixel_buf_dev
    alt_up_pixel_buffer_dma_open_dev("/dev/VGA_Pixel_Buffer");

borra_captura();
while(1)
{
    clear_screen(pixel_buf_dev);
    sw =(pSWITCH) & 0x02;
    GPIO0 =(pGPIO0);
    EOC=(pEOC);
    while(sw==2) //pregunto si no esta reseteando el software.
    {
        VGA_text (70, 15, text_top_row);
        captura_amp();
        VGA_text(70,25,amp_char);//floor(*(poffset)*39.0625)
        VGA_text (5, 25, "EjeY:");
        VGA_text (2, 27, "1 div = 1 V");
        VGA_text (25, 1,"TESIS OSCILOSCOPIO: VALVERDE-RUIZ");
    }
}

```

```

VGA_text (65, 29, "0");
VGA_text (70, 35, voltaje);
captura_offset();
VGA_text (70, 37, "                ");
VGA_text(70,37,V_offset_char);//floor(*(poffset)*39.0625)
frec =*(pSWITCH) & 0x7000;
switch ( frec ) {
    case 28672: VGA_text (14, 57, "Eje X: 1 div = 2,4625 ms
"); break;
    case 4096: VGA_text (14, 57, "Eje X: 1 div = 26,8 ms
"); break;
    case 8192: VGA_text (14, 57, "Eje X: 1 div = 13,6 ms
"); break;
    case 12288: VGA_text (14, 57, "Eje X: 1 div = 7 ms
"); break;
    case 16384: VGA_text (14, 57, "Eje X: 1 div = 4,8 ms
"); break;
    case 20480: VGA_text (14, 57, "Eje X: 1 div = 3,7 ms
"); break;
    case 24576: VGA_text (14, 57, "Eje X: 1 div = 3,04 ms
"); break;
    default: VGA_text (14, 57, "Eje X: 1 div = 132,41 ms
"); break;
}
strcpy(modos, "MODO: MUESTREO CONTINUO\0");
VGA_text (43, 57, modos);
i=0;
while( i<=199)           //Capturo Simbolo
{
    EOC=*(pEOC) & 0x01;
    if (EOC==1)
    {
        captura_dato();
        V[i]=dato;
        EOC=*(pEOC) & 0x01;
        while(EOC==1)           // espero a q dato listo cambie
        {
            EOC=*(pEOC) & 0x01;

```

```

        }
        i++;
    }
}
for(i=0; i<=199;i++)//borro simbolo anterior
{
    x=i+59;
    if(i>0)

alt_up_pixel_buffer_dma_draw_line(pixel_buf_dev,x-
1,V_ant[i-1],x,V_ant[i],0,0);
}
crear_cuadrricula( pixel_buf_dev);
for(i=0; i<=199;i++) //imprimo simbolo actual
{
    x=i+59;
    if(i>0)

alt_up_pixel_buffer_dma_draw_line(pixel_buf_dev,x-1,V[i-
1],x,V[i],0xFFFF,0);
    V_ant[i]=V[i]; //Convierto V en V_ant
}
    stop=*(pSWITCH) & 0x40;
    while (stop==64)
    { stop=*(pSWITCH) & 0x40;
      strcpy(modo, "MOD0: FOTOGRAFIA \0");
      VGA_text (43, 57, modo);
    }
    sw =*(pSWITCH) & 0x02;
}
}
}

```

```

//***** V G A *****

void VGA_text(int x, int y, char * text_ptr)
{
    int offset;
    volatile char * character_buffer = (char *) 0x01080000; //
    VGA character buffer
    offset = (y << 7) + x;
    while ( *(text_ptr) )
    {
        *(character_buffer + offset) = *(text_ptr); //
        escribo en el buffer de caracteres
        ++text_ptr;
        ++offset;
    }
}

void VGA_box(int x1, int s1, int x2, int y2, short pixel_color)
{
    int offset, row, col;
    volatile short * pixel_buffer = (short *) 0x01000000; // VGA
    buffer de pixeles
    for (row = s1; row <= y2; row++)
    {
        col = x1;
        while (col <= x2)
        {
            offset = (row << 9) + col;
            *(pixel_buffer + offset) = pixel_color;
            ++col;
        }
    }
}

int cargo_dato(int dato, int V[i])
{
    int i;
}

```

```

        for(i=0; i<=199;i++)
        {
            if(i<=199)
                {V[i]=V[i+1];
                }else{
                V[i]= dato;
                }
        }
    }

void write_pixel(int x, int y, int colour)
{
    volatile short *vga_addr=(volatile short*)(0x01000000 + (y<<10) +
(x<<1));
    *vga_addr=colour;
}

void clear_screen(alt_up_pixel_buffer_dma_dev *pixel_buffer)
{
    alt_up_pixel_buffer_dma_clear_screen(pixel_buffer,0);
}

void crear_cuadrícula(alt_up_pixel_buffer_dma_dev *pixel_buffer)
{
    int x,y;
    for (x = 59; x <260 ; x=x+20)
    {
        alt_up_pixel_buffer_dma_draw_line(pixel_buffer,x,19,x,220,0x18
7F,0);
    }
    for (y = 19; y <220 ; y=y+20)
    {
        alt_up_pixel_buffer_dma_draw_line(pixel_buffer,59,y,260,y,0x18
7F,0);
    }
}

```

```

void mostrar(int V[j], int desplazamiento)
{
    int j;
    for (j = 259; j<59 ; j=j--)
    {
        write_pixel(j,V[j],0x187F);
    }
}

void borra_captura()
{
    for (i=0; i<=280; i=i++)
        {V_ant[i]=119;
    }
}

int captura_dato()
{
    int dato0, dato1, dato2, dato3, dato4, dato5, dato6, dato7;
    volatile int *pGPIO0= (int*)0x010830e0;
    float dato_float;

    int GPIO0 = *pGPIO0;
    dato0 = GPIO0 & 0x01;
    dato1 = GPIO0 & 0x02;
    dato2 = GPIO0 & 0x04;
    dato3 = GPIO0 & 0x08;
    dato4 = GPIO0 & 0x10;
    dato5 = GPIO0 & 0x20;
    dato6 = GPIO0 & 0x40;
    dato7 = GPIO0 & 0x80;

    dato_float = dato0 + dato1 + dato2 + dato3 + dato4 +
dato5 + dato6 + dato7;
    dato = 219-(int)floor(dato_float*200/256);
}

```



```

int captura_offset()
{
    int dato0, dato1, dato2, dato3, dato4, dato5, dato6, dato7;
    volatile int *poffset= (int*)0x00000000;
    int GPIO0 = *poffset;
    float v_offset_float;

    dato0 = GPIO0 & 0x01;
    dato1 = GPIO0 & 0x02;
    dato2 = GPIO0 & 0x04;
    dato3 = GPIO0 & 0x08;
    dato4 = GPIO0 & 0x10;
    dato5 = GPIO0 & 0x20;
    dato6 = GPIO0 & 0x40;
    dato7 = GPIO0 & 0x80;

    v_offset_float = (float) (dato0 + dato1 + dato2 + dato3 +
    dato4 + dato5 + dato6 + dato7);

    v_offset_float = (115-v_offset_float)*10/256; //en las
    pruebas de calibracion se observo que el offset nulo era 115
    //offset debe ir negado porq el circuito electronico lo
    resta.
    sprintf(V_offset_char,"%g", v_offset_float);
}

int captura_amp()
{
    int dato0, dato1, dato2, RF;
    volatile int *pRF= (int*)0x00000010;
    int GPIO0 = *pRF;

    dato0 = GPIO0 & 0x01;
    dato1 = GPIO0 & 0x02;
    dato2 = GPIO0 & 0x04;

    RF = dato0 + dato1 + dato2;

```

```
switch ( RF ) {
case 0:
    strcpy(amp_char, "AMP: 1X \0");
    break;
case 1:
    strcpy(amp_char, "AMP: 2X \0");
    break;
case 2:
    strcpy(amp_char, "AMP: 3X \0");
    break;
case 3:
    strcpy(amp_char, "AMP: 4.7X \0");
    break;
case 4:
    strcpy(amp_char, "AMP: 10X \0");
    break;
case 5:
    strcpy(amp_char, "AMP: 15X \0");
    break;
case 6:
    strcpy(amp_char, "AMP: 47X \0");
    break;
default:
    strcpy(amp_char, "AMP: 100X \0");
    break;
}
}
```

ANEXO B

```
void main()
{
    volatile int *pKEY = (int*)0x1083070;
    volatile int *pSWITCH= (int*)0x1083060;
    volatile int *pGPIO0= (int*)0x010830e0;
    volatile int *pEOC= (int*)0x010830f0;
    volatile int *pLEDR = (int*)0x01083020;
    volatile int *vga_addr;
    volatile int *poffset= (int*)0x00000000;
    volatile int *pRF= (int*)0x00000010;
    int sw,GPIO0,EOC,LEDR, datolis, stop, frec;

    alt_up_pixel_buffer_dma_dev *pixel_buf_dev;

    pixel_buf_dev=alt_up_pixel_buffer_dma_open_dev("/dev/VGA_Pixel
_Buffer");

    borra_captura();

    while(1)
    {
        clear_screen(pixel_buf_dev);
        sw =(pSWITCH) & 0x02;
        GPIO0 =(pGPIO0);
        EOC=(pEOC);
        while(sw==2)//pregunto si no esta reseteando el
software.
        {
            VGA_text (70, 15, text_top_row);
            captura_amp();
            VGA_text (70,25,amp_char);//floor>(* (poffset)*39.0625)
```

```

    VGA_text (5, 25, "EjeY:");
    VGA_text (2, 27, "1 div = 1 V");
    VGA_text (25, 1, "TESIS OSCILOSCOPIO: VALVERDE-RUIZ");
    VGA_text (70, 35, voltaje);
    captura_offset();
    VGA_text (70, 37, "
");
VGA_text(70,37,V_offset_char);//floor(*(poffset)*39.0625)

frec =*(pSWITCH) & 0x7000;
switch ( frec ) {
case 28672: VGA_text (14, 57, "Eje X: 1 div = 2,4625 ms
"); break;
case 4096: VGA_text (14, 57, "Eje X: 1 div = 26,8 ms
"); break;
case 8192: VGA_text (14, 57, "Eje X: 1 div = 13,6 ms
"); break;
case 12288: VGA_text (14, 57, "Eje X: 1 div = 7 ms
"); break;
case 16384: VGA_text (14, 57, "Eje X: 1 div = 4,8 ms
"); break;
case 20480: VGA_text (14, 57, "Eje X: 1 div = 3,7 ms
"); break;
case 24576: VGA_text (14, 57, "Eje X: 1 div = 3,04 ms
"); break;
default: VGA_text (14, 57, "Eje X: 1 div = 132,41 ms
"); break;
}

strcpy(modos, "MODO: MUESTREO CONTINUO\0");
VGA_text (43, 57, modos);
i=0;
while( i<=199) //Capturo Simbolo
{
    EOC=*(pEOC) & 0x01;
    if (EOC==1)
    {
        DATO_8BITS=captura_dato();
        V[i]=dato;
    }
}

```

```

        EOC=*(pEOC) & 0x01;
        while(EOC==1)           // espero a q dato listo cambie
        {
            EOC=*(pEOC) & 0x01;
        }
        i++;
    }
    for(i=0; i<=199;i++)        //borro simbolo anterior
    {
        x=i+59;
        if(i>0)
            alt_up_pixel_buffer_dma_draw_line(pixel_buf_dev,x-
            1,V_ant[i-1],x,V_ant[i],0,0);
    }
    crear_cuadrricula( pixel_buf_dev);
    for(i=0; i<=199;i++)        //imprimo simbolo actual
    {
        x=i+59;
        if(i>0)
            alt_up_pixel_buffer_dma_draw_line(pixel_buf_dev,x-1,V[i-
            1],x,V[i],0xFFFF,0);
        V_ant[i]=V[i];          //Convierto V en V_ant
    }
//Codigo agregado para las pruebas
    VGA_text (2, 29, "Dato-decimal:");
    sprintf(Dato_char,"%g", DATO_8BITS);
    VGA_text (2, 30, "          ");
    VGA_text (2, 30, Dato_char);
    VGA_text (2, 32, "Dato-Voltios:");
    VGA_text (2, 33, "          ");
    sprintf(Dato_char,"%g", DATO_8BITS*10/255-5);
    VGA_text (2, 33, Dato_char);
    stop=*(pSWITCH) & 0x40;
    while (stop==64)
    { stop=*(pSWITCH) & 0x40;
        strcpy(modos, "MODO: FOTOGRAFIA \0");
    }

```

```
        VGA_text (43, 57, modo);
    }
    sw =*(pSWITCH) & 0x02;
}
}
```

ANEXO C

VARIACIÓN DEL VOLTAJE OFFSET DIGITAL

OFFSET	VALOR TEORICO	VALOR MEDIDO	PASO DEL OFFSET	ERROR	ERROR DE PASO
10	4,1015625	4,13725	-	0,87%	-
11	4,0625	4,09804	0,03921	0,87%	0,38%
12	4,0234375	4,05882	0,03922	0,88%	0,40%
13	3,984375	4,01961	0,03921	0,88%	0,38%
14	3,9453125	3,98039	0,03922	0,89%	0,40%
15	3,90625	3,94118	0,03921	0,89%	0,38%
16	3,8671875	3,90196	0,03922	0,90%	0,40%
17	3,828125	3,86275	0,03921	0,90%	0,38%
18	3,7890625	3,82353	0,03922	0,91%	0,40%
19	3,75	3,78431	0,03922	0,91%	0,40%
20	3,7109375	3,7451	0,03921	0,92%	0,38%
21	3,671875	3,70588	0,03922	0,93%	0,40%
22	3,6328125	3,66667	0,03921	0,93%	0,38%
23	3,59375	3,62745	0,03922	0,94%	0,40%
24	3,5546875	3,58823	0,03922	0,94%	0,40%
25	3,515625	3,54902	0,03921	0,95%	0,38%
26	3,4765625	3,5098	0,03922	0,96%	0,40%
27	3,4375	3,47059	0,03921	0,96%	0,38%
28	3,3984375	3,43137	0,03922	0,97%	0,40%
29	3,359375	3,39216	0,03921	0,98%	0,38%
30	3,3203125	3,35294	0,03922	0,98%	0,40%
31	3,28125	3,31373	0,03921	0,99%	0,38%
32	3,2421875	3,27451	0,03922	1,00%	0,40%
33	3,203125	3,23529	0,03922	1,00%	0,40%
34	3,1640625	3,19608	0,03921	1,01%	0,38%
35	3,125	3,15686	0,03922	1,02%	0,40%
36	3,0859375	3,11765	0,03921	1,03%	0,38%
37	3,046875	3,07843	0,03922	1,04%	0,40%

38	3,0078125	3,03922	0,03921	1,04%	0,38%
39	2,96875	3	0,03922	1,05%	0,40%
40	2,9296875	2,96078	0,03922	1,06%	0,40%
41	2,890625	2,92157	0,03921	1,07%	0,38%
42	2,8515625	2,88235	0,03922	1,08%	0,40%
43	2,8125	2,84314	0,03921	1,09%	0,38%
44	2,7734375	2,80392	0,03922	1,10%	0,40%
45	2,734375	2,76471	0,03921	1,11%	0,38%
46	2,6953125	2,72549	0,03922	1,12%	0,40%
47	2,65625	2,68627	0,03922	1,13%	0,40%
48	2,6171875	2,64706	0,03921	1,14%	0,38%
49	2,578125	2,60784	0,03922	1,15%	0,40%
50	2,5390625	2,56863	0,03921	1,16%	0,38%
51	2,5	2,52941	0,03922	1,18%	0,40%
52	2,4609375	2,4902	0,03921	1,19%	0,38%
53	2,421875	2,45098	0,03922	1,20%	0,40%
54	2,3828125	2,41176	0,03922	1,21%	0,40%
55	2,34375	2,37255	0,03921	1,23%	0,38%
56	2,3046875	2,33333	0,03922	1,24%	0,40%
57	2,265625	2,29412	0,03921	1,26%	0,38%
58	2,2265625	2,2549	0,03922	1,27%	0,40%
59	2,1875	2,21569	0,03921	1,29%	0,38%
60	2,1484375	2,17647	0,03922	1,30%	0,40%
61	2,109375	2,13725	0,03922	1,32%	0,40%
62	2,0703125	2,09804	0,03921	1,34%	0,38%
63	2,03125	2,05882	0,03922	1,36%	0,40%
64	1,9921875	2,01961	0,03921	1,38%	0,38%
65	1,953125	1,98039	0,03922	1,40%	0,40%
66	1,9140625	1,94118	0,03921	1,42%	0,38%
67	1,875	1,90196	0,03922	1,44%	0,40%
68	1,8359375	1,86275	0,03921	1,46%	0,38%
69	1,796875	1,82353	0,03922	1,48%	0,40%
70	1,7578125	1,78431	0,03922	1,51%	0,40%
71	1,71875	1,7451	0,03921	1,53%	0,38%
72	1,6796875	1,70588	0,03922	1,56%	0,40%
73	1,640625	1,66667	0,03921	1,59%	0,38%

74	1,6015625	1,62745	0,03922	1,62%	0,40%
75	1,5625	1,58824	0,03921	1,65%	0,38%
76	1,5234375	1,54902	0,03922	1,68%	0,40%
77	1,484375	1,5098	0,03922	1,71%	0,40%
78	1,4453125	1,47059	0,03921	1,75%	0,38%
79	1,40625	1,43137	0,03922	1,79%	0,40%
80	1,3671875	1,39216	0,03921	1,83%	0,38%
81	1,328125	1,35294	0,03922	1,87%	0,40%
82	1,2890625	1,31373	0,03921	1,91%	0,38%
83	1,25	1,27451	0,03922	1,96%	0,40%
84	1,2109375	1,23529	0,03922	2,01%	0,40%
85	1,171875	1,19608	0,03921	2,07%	0,38%
86	1,1328125	1,15686	0,03922	2,12%	0,40%
87	1,09375	1,11765	0,03921	2,19%	0,38%
88	1,0546875	1,07843	0,03922	2,25%	0,40%
89	1,015625	1,03922	0,03921	2,32%	0,38%
90	0,9765625	1	0,03922	2,40%	0,40%
91	0,9375	0,960784	0,039216	2,48%	0,39%
92	0,8984375	0,921568	0,039216	2,57%	0,39%
93	0,859375	0,882353	0,039215	2,67%	0,39%
94	0,8203125	0,843137	0,039216	2,78%	0,39%
95	0,78125	0,803922	0,039215	2,90%	0,39%
96	0,7421875	0,764706	0,039216	3,03%	0,39%
97	0,703125	0,725449	0,039257	3,17%	0,50%
98	0,6640625	0,686275	0,039174	3,34%	0,29%
99	0,625	0,647059	0,039216	3,53%	0,39%
100	0,5859375	0,607843	0,039216	3,74%	0,39%
101	0,546875	0,568627	0,039216	3,98%	0,39%
102	0,5078125	0,529412	0,039215	4,25%	0,39%
103	0,46875	0,490196	0,039216	4,58%	0,39%
104	0,4296875	0,45098	0,039216	4,96%	0,39%
105	0,390625	0,411765	0,039215	5,41%	0,39%
106	0,3515625	0,372549	0,039216	5,97%	0,39%
107	0,3125	0,333333	0,039216	6,67%	0,39%
108	0,2734375	0,294117	0,039216	7,56%	0,39%
109	0,234375	0,254902	0,039215	8,76%	0,39%

110	0,1953125	0,215686	0,039216	10,43%	0,39%
111	0,15625	0,176471	0,039215	12,94%	0,39%
112	0,1171875	0,137255	0,039216	17,12%	0,39%
113	0,078125	0,0980392	0,0392158	25,49%	0,39%
114	0,0390625	0,0588236	0,0392156	50,59%	0,39%
115	0	0,019608	0,0392156	-	0,39%
116	-0,0390625	-0,019608	0,039216	49,80%	0,39%
117	-0,078125	-0,0588239	0,0392159	24,71%	0,39%
118	-0,1171875	-0,0980392	0,0392153	16,34%	0,39%
119	-0,15625	-0,137255	0,0392158	12,16%	0,39%
120	-0,1953125	-0,176471	0,039216	9,65%	0,39%
121	-0,234375	-0,215686	0,039215	7,97%	0,39%
122	-0,2734375	-0,254902	0,039216	6,78%	0,39%
123	-0,3125	-0,294117	0,039215	5,88%	0,39%
124	-0,3515625	-0,333333	0,039216	5,19%	0,39%
125	-0,390625	-0,372549	0,039216	4,63%	0,39%
126	-0,4296875	-0,411765	0,039216	4,17%	0,39%
127	-0,46875	-0,45098	0,039215	3,79%	0,39%
128	-0,5078125	-0,490196	0,039216	3,47%	0,39%
129	-0,546875	-0,529412	0,039216	3,19%	0,39%
130	-0,5859375	-0,568627	0,039215	2,95%	0,39%
131	-0,625	-0,607843	0,039216	2,75%	0,39%
132	-0,6640625	-0,647059	0,039216	2,56%	0,39%
133	-0,703125	-0,686275	0,039216	2,40%	0,39%
134	-0,7421875	-0,72549	0,039215	2,25%	0,39%
135	-0,78125	-0,764706	0,039216	2,12%	0,39%
136	-0,8203125	-0,803922	0,039216	2,00%	0,39%
137	-0,859375	-0,843137	0,039215	1,89%	0,39%
138	-0,8984375	-0,882353	0,039216	1,79%	0,39%
139	-0,9375	-0,921569	0,039216	1,70%	0,39%
140	-0,9765625	-0,960784	0,039215	1,62%	0,39%
141	-1,015625	-1	0,039216	1,54%	0,39%
142	-1,0546875	-1,03922	0,03922	1,47%	0,40%
143	-1,09375	-1,07843	0,03921	1,40%	0,38%
144	-1,1328125	-1,11765	0,03922	1,34%	0,40%
145	-1,171875	-1,15686	0,03921	1,28%	0,38%

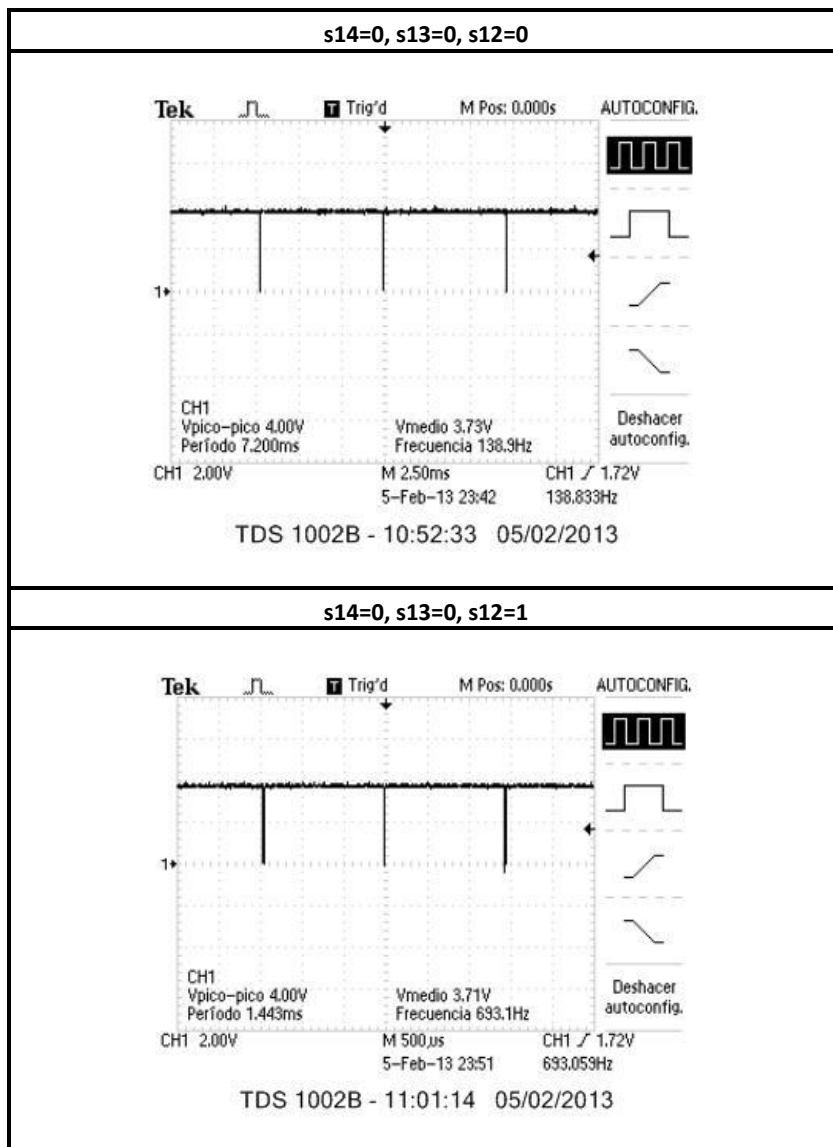
146	-1,2109375	-1,19608	0,03922	1,23%	0,40%
147	-1,25	-1,23529	0,03921	1,18%	0,38%
148	-1,2890625	-1,27451	0,03922	1,13%	0,40%
149	-1,328125	-1,31373	0,03922	1,08%	0,40%
150	-1,3671875	-1,35294	0,03921	1,04%	0,38%
151	-1,40625	-1,39216	0,03922	1,00%	0,40%
152	-1,4453125	-1,43137	0,03921	0,96%	0,38%
153	-1,484375	-1,47059	0,03922	0,93%	0,40%
154	-1,5234375	-1,5098	0,03921	0,90%	0,38%
155	-1,5625	-1,54902	0,03922	0,86%	0,40%
156	-1,6015625	-1,58824	0,03922	0,83%	0,40%
157	-1,640625	-1,62745	0,03921	0,80%	0,38%
158	-1,6796875	-1,66667	0,03922	0,77%	0,40%
159	-1,71875	-1,70588	0,03921	0,75%	0,38%
160	-1,7578125	-1,7451	0,03922	0,72%	0,40%
161	-1,796875	-1,78431	0,03921	0,70%	0,38%
162	-1,8359375	-1,82353	0,03922	0,68%	0,40%
163	-1,875	-1,86275	0,03922	0,65%	0,40%
164	-1,9140625	-1,90196	0,03921	0,63%	0,38%
165	-1,953125	-1,94118	0,03922	0,61%	0,40%
166	-1,9921875	-1,98039	0,03921	0,59%	0,38%
167	-2,03125	-2,01961	0,03922	0,57%	0,40%
168	-2,0703125	-2,05882	0,03921	0,56%	0,38%
169	-2,109375	-2,09804	0,03922	0,54%	0,40%
170	-2,1484375	-2,13725	0,03921	0,52%	0,38%
171	-2,1875	-2,17647	0,03922	0,50%	0,40%
172	-2,2265625	-2,21569	0,03922	1,27%	0,40%
173	-2,265625	-2,2549	0,03921	1,26%	0,38%
174	-2,3046875	-2,29412	0,03922	1,24%	0,40%
175	-2,34375	-2,33333	0,03921	1,23%	0,38%
176	-2,3828125	-2,37255	0,03922	1,21%	0,40%
177	-2,421875	-2,41176	0,03921	1,20%	0,38%
178	-2,4609375	-2,45098	0,03922	1,19%	0,40%
179	-2,5	-2,4902	0,03922	1,18%	0,40%
180	-2,5390625	-2,52941	0,03921	1,16%	0,38%
181	-2,578125	-2,56863	0,03922	1,15%	0,40%

182	-2,6171875	-2,60784	0,03921	1,14%	0,38%
183	-2,65625	-2,64706	0,03922	1,13%	0,40%
184	-2,6953125	-2,68627	0,03921	1,12%	0,38%
185	-2,734375	-2,72549	0,03922	1,11%	0,40%
186	-2,7734375	-2,76471	0,03922	1,10%	0,40%
187	-2,8125	-2,80392	0,03921	1,09%	0,38%
188	-2,8515625	-2,84314	0,03922	1,08%	0,40%
189	-2,890625	-2,88235	0,03921	1,07%	0,38%
190	-2,9296875	-2,92157	0,03922	1,06%	0,40%
191	-2,96875	-2,96078	0,03921	1,05%	0,38%
192	-3,0078125	-3	0,03922	1,04%	0,40%
193	-3,046875	-3,03922	0,03922	1,04%	0,40%
194	-3,0859375	-3,07843	0,03921	1,03%	0,38%
195	-3,125	-3,11765	0,03922	1,02%	0,40%
196	-3,1640625	-3,15686	0,03921	1,01%	0,38%
197	-3,203125	-3,19608	0,03922	1,00%	0,40%
198	-3,2421875	-3,23529	0,03921	1,00%	0,38%
199	-3,28125	-3,27451	0,03922	0,99%	0,40%
200	-3,3203125	-3,31373	0,03922	0,98%	0,40%
201	-3,359375	-3,35294	0,03921	0,98%	0,38%
202	-3,3984375	-3,39216	0,03922	0,97%	0,40%
203	-3,4375	-3,43137	0,03921	0,96%	0,38%
204	-3,4765625	-3,47059	0,03922	0,96%	0,40%
205	-3,515625	-3,5098	0,03921	0,95%	0,38%
206	-3,5546875	-3,54902	0,03922	0,94%	0,40%
207	-3,59375	-3,58824	0,03922	0,94%	0,40%
208	-3,6328125	-3,62745	0,03921	0,93%	0,38%
209	-3,671875	-3,66667	0,03922	0,93%	0,40%
210	-3,7109375	-3,70588	0,03921	0,92%	0,38%
211	-3,75	-3,7451	0,03922	0,91%	0,40%
212	-3,7890625	-3,78431	0,03921	0,91%	0,38%
213	-3,828125	-3,82353	0,03922	0,90%	0,40%
214	-3,8671875	-3,86275	0,03922	0,90%	0,40%
215	-3,90625	-3,90196	0,03921	0,89%	0,38%
216	-3,9453125	-3,94118	0,03922	0,89%	0,40%
217	-3,984375	-3,98039	0,03921	0,88%	0,38%

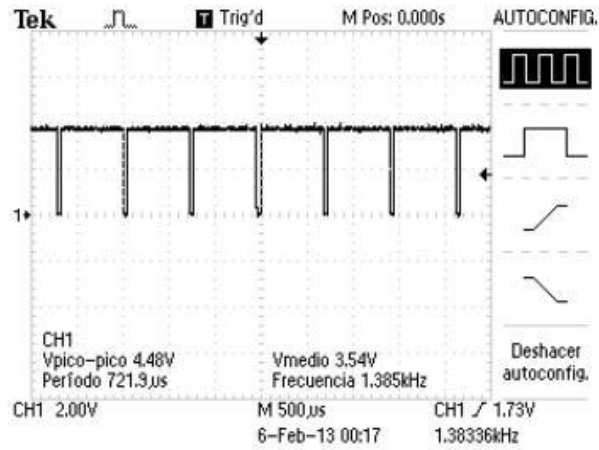
218	-4,0234375	-4,01961	0,03922	0,88%	0,40%
219	-4,0625	-4,05882	0,03921	0,87%	0,38%
220	-4,1015625	-4,09804	0,03922	0,87%	0,40%
221	-4,140625	-4,13725	0,03921	0,87%	0,38%
222	-4,1796875	-4,17647	0,03922	0,86%	0,40%
223	-4,21875	-4,21569	0,03922	0,86%	0,40%
224	-4,2578125	-4,2549	0,03921	0,85%	0,38%
225	-4,296875	-4,29412	0,03922	0,85%	0,40%
226	-4,3359375	-4,33333	0,03921	0,84%	0,38%
227	-4,375	-4,37255	0,03922	0,84%	0,40%
228	-4,4140625	-4,41176	0,03921	0,84%	0,38%
229	-4,453125	-4,45098	0,03922	0,83%	0,40%
230	-4,4921875	-4,4902	0,03922	0,83%	0,40%
231	-4,53125	-4,52941	0,03921	0,82%	0,38%
232	-4,5703125	-4,56863	0,03922	0,82%	0,40%
233	-4,609375	-4,60784	0,03921	0,82%	0,38%
234	-4,6484375	-4,64706	0,03922	0,81%	0,40%
235	-4,6875	-4,68627	0,03921	0,81%	0,38%
236	-4,7265625	-4,72549	0,03922	0,81%	0,40%
237	-4,765625	-4,76471	0,03922	0,80%	0,40%
238	-4,8046875	-4,80392	0,03921	0,80%	0,38%
239	-4,84375	-4,84314	0,03922	0,80%	0,40%
240	-4,8828125	-4,88235	0,03921	0,79%	0,38%
241	-4,921875	-4,92157	0,03922	0,79%	0,40%
242	-4,9609375	-4,96078	0,03921	0,79%	0,38%
243	-5	-5	0,03922	-	-

ANEXO D

MEDICION DE FRECUENCIA DE MUESTREO

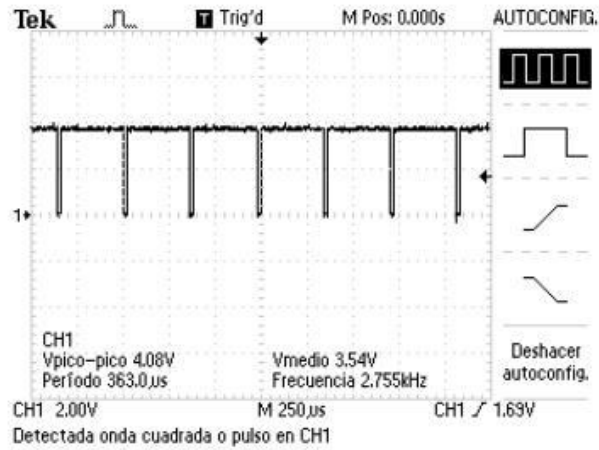


s14=0, s13=1, s12=0



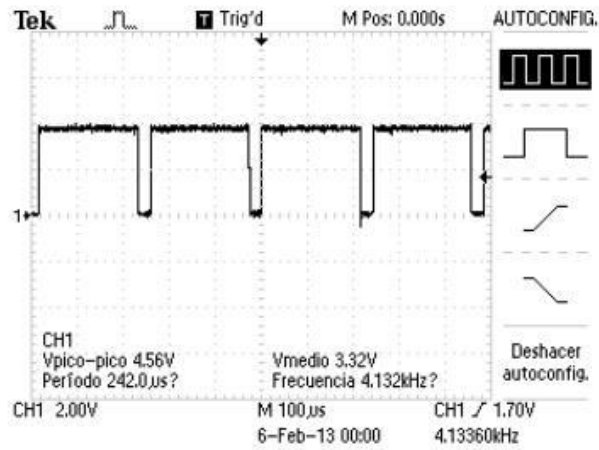
TDS 1002B - 11:28:04 05/02/2013

s14=0, s13=1, s12=1



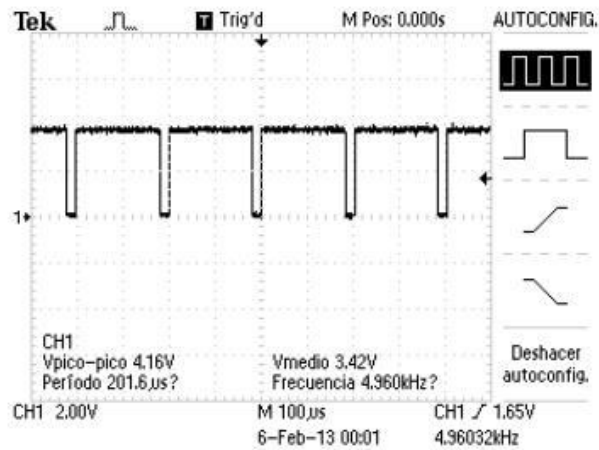
TDS 1002B - 11:05:52 05/02/2013

s14=1, s13=0, s12=0



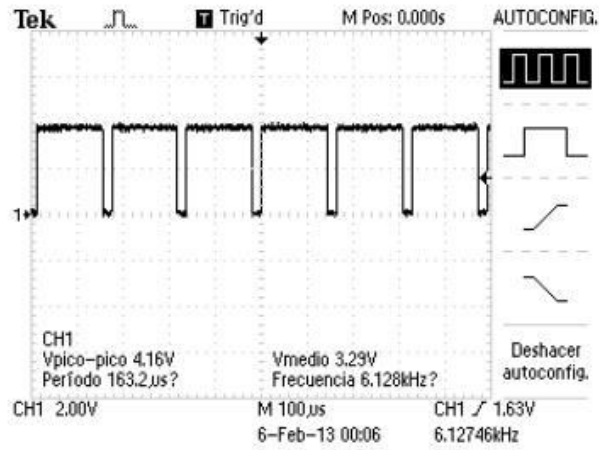
TDS 1002B - 11:10:30 05/02/2013

s14=1, s13=0, s12=1



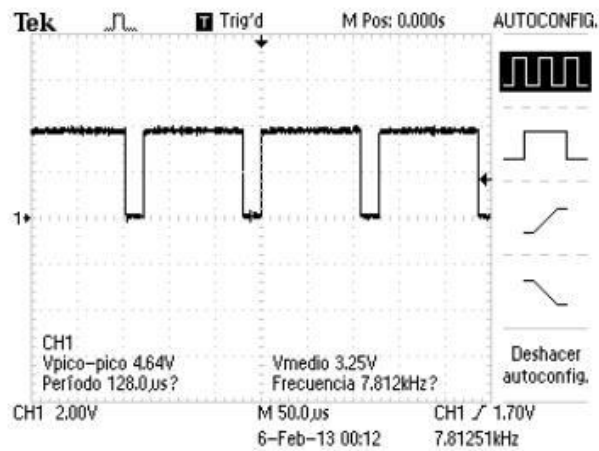
TDS 1002B - 11:12:05 05/02/2013

s14=1, s13=1, s12=0



TDS 1002B - 11:16:17 05/02/2013

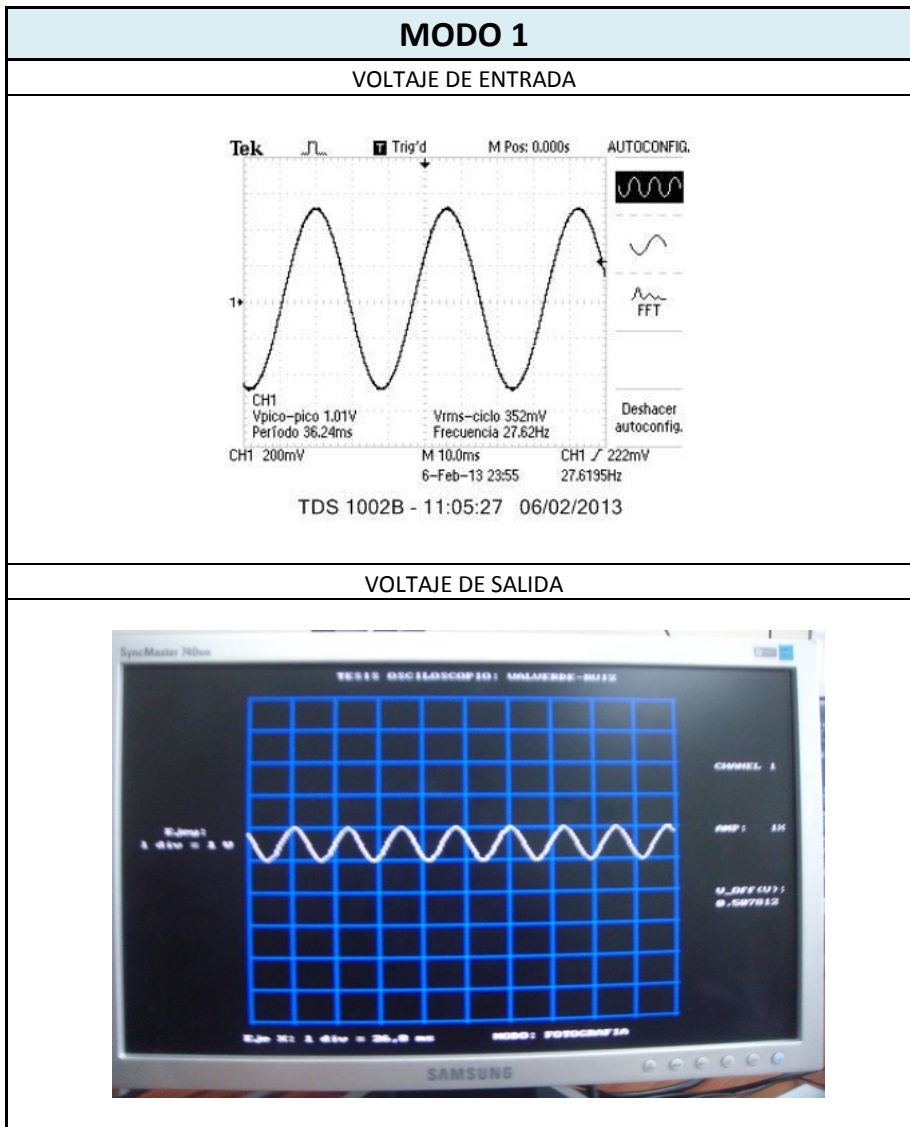
s14=1, s13=1, s12=1



TDS 1002B - 11:22:19 05/02/2013

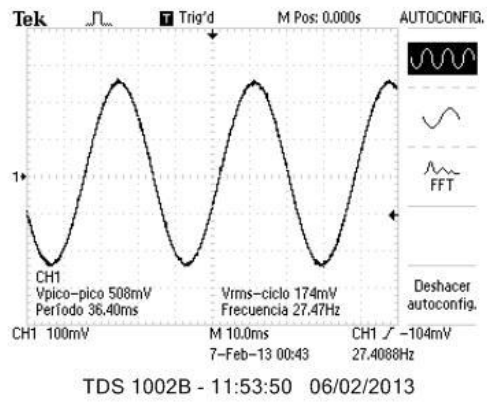
ANEXO E

MEDICION DE FACTORES DE AMPLIFICACION

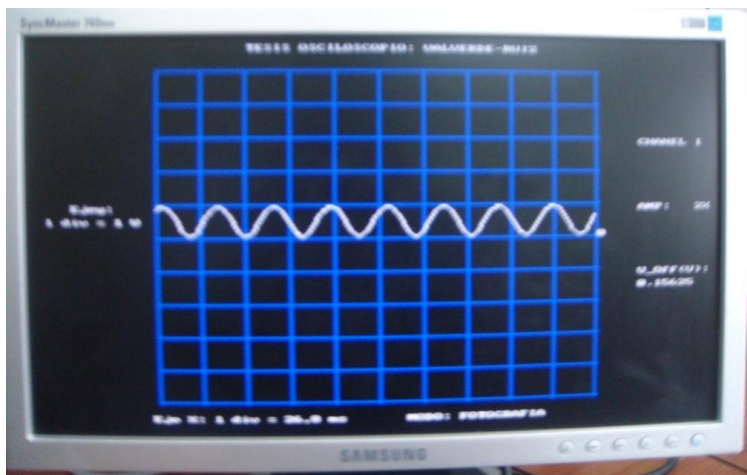


MODO 2

VOLTAJE DE ENTRADA

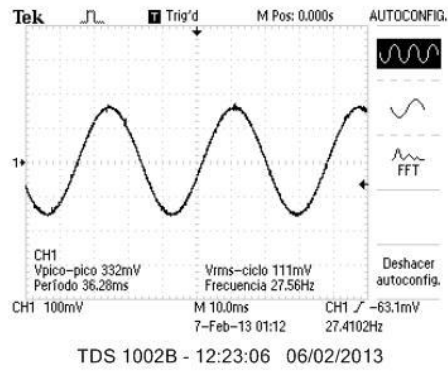


VOLTAJE DE SALIDA

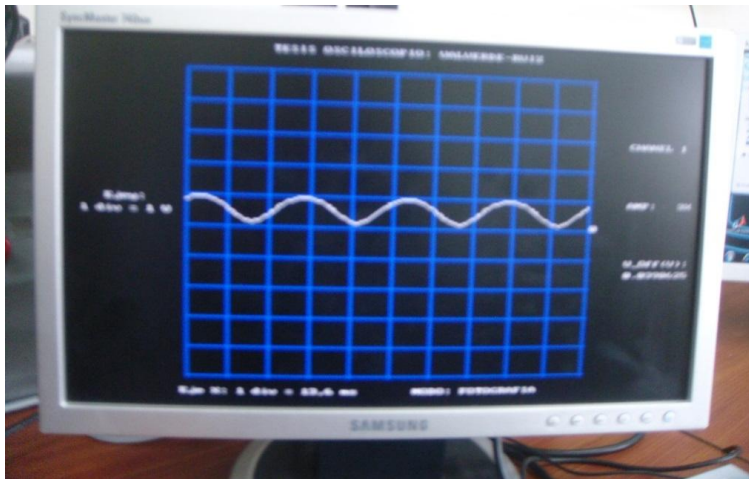


MODO 3

VOLTAJE DE ENTRADA

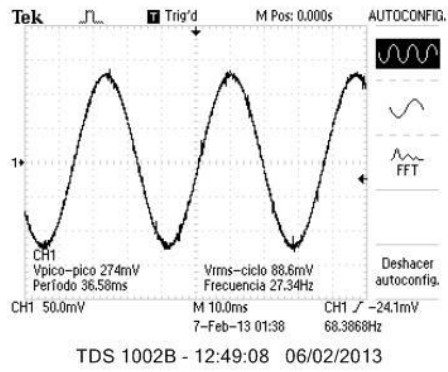


VOLTAJE DE SALIDA

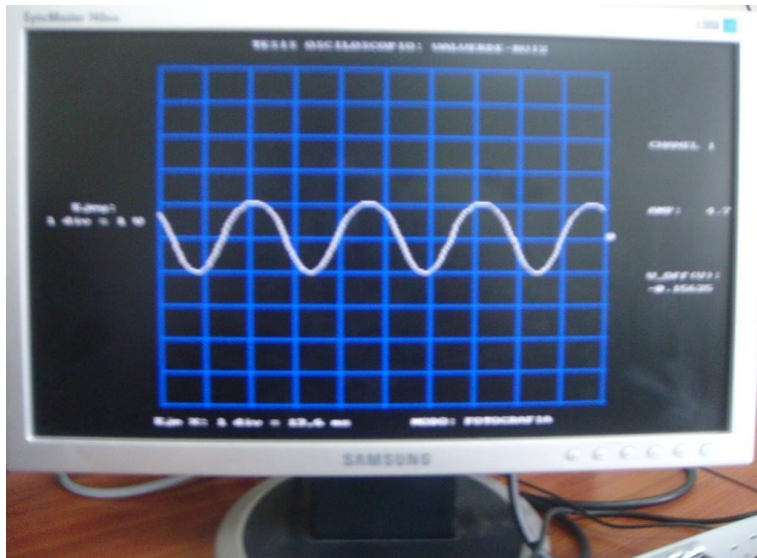


MODO 4

VOLTAJE DE ENTRADA



VOLTAJE DE SALIDA



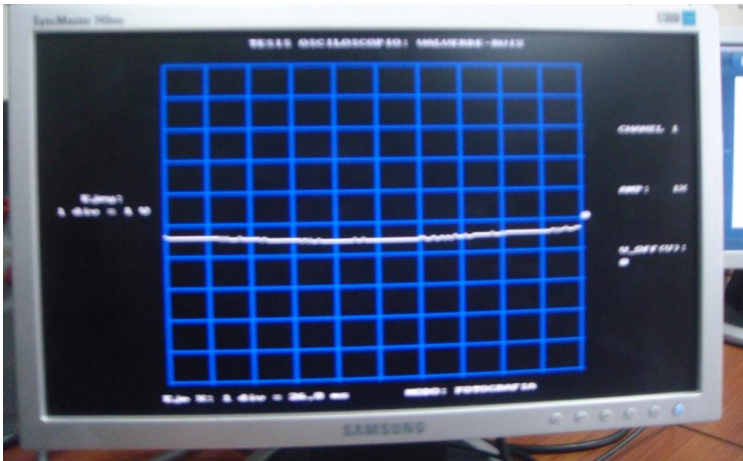
ANEXO F

VARIACION DE LA FRECUENCIA DE LA SEÑAL

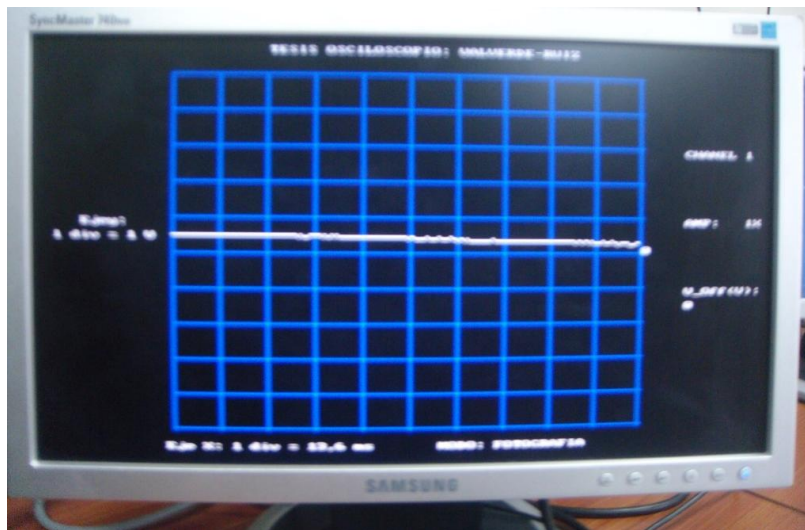
Frecuencia de Señal = 1 Hz, Frecuencia de Muestreo = 138,7 Hz



Frecuencia de Señal = 1 Hz, Frecuencia de Muestreo = 693,1 Hz



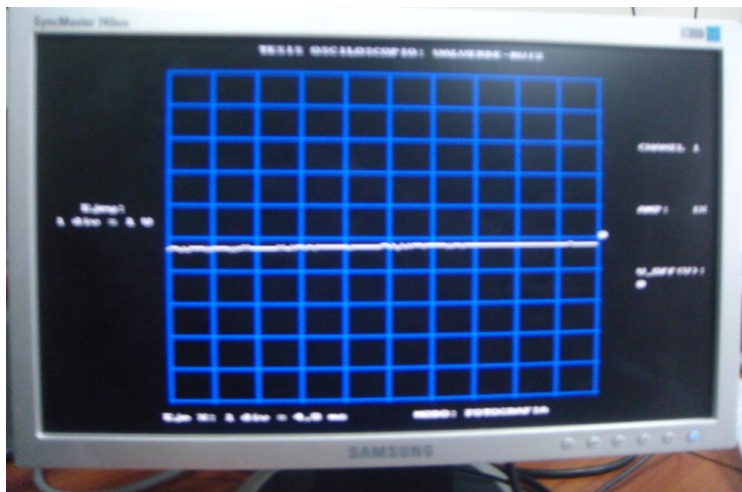
Frecuencia de Señal = 1 Hz, Frecuencia de Muestreo = 1383 Hz



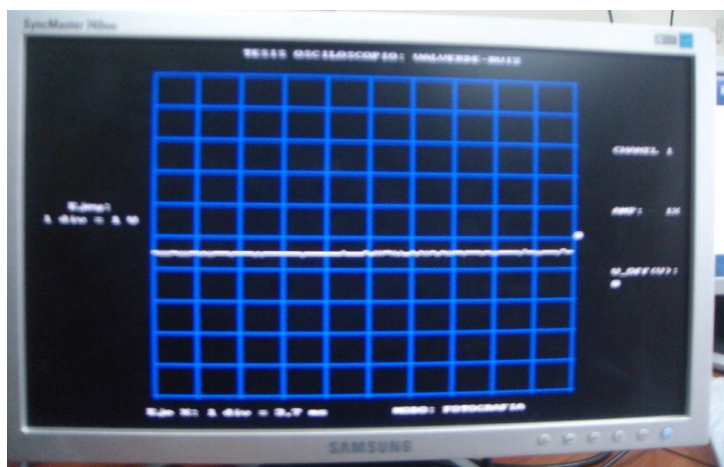
Frecuencia de Señal = 1 Hz, Frecuencia de Muestreo = 2755 Hz



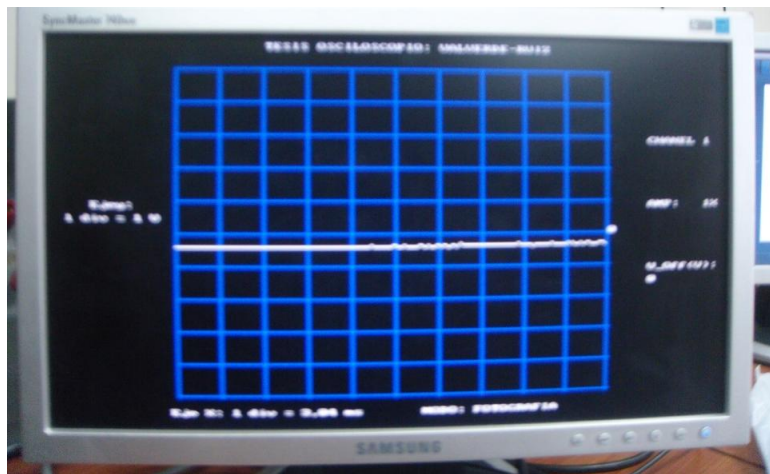
Frecuencia de Señal = 1 Hz, Frecuencia de Muestreo = 4132 Hz



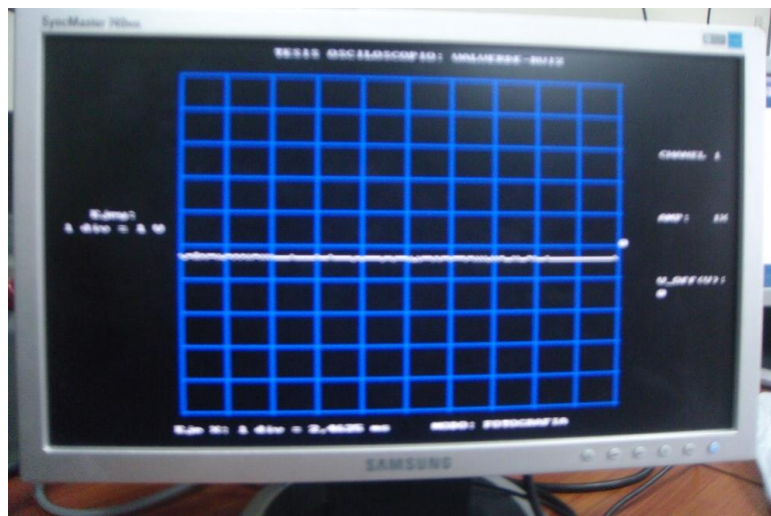
Frecuencia de Señal = 1 Hz, Frecuencia de Muestreo = 4975 Hz



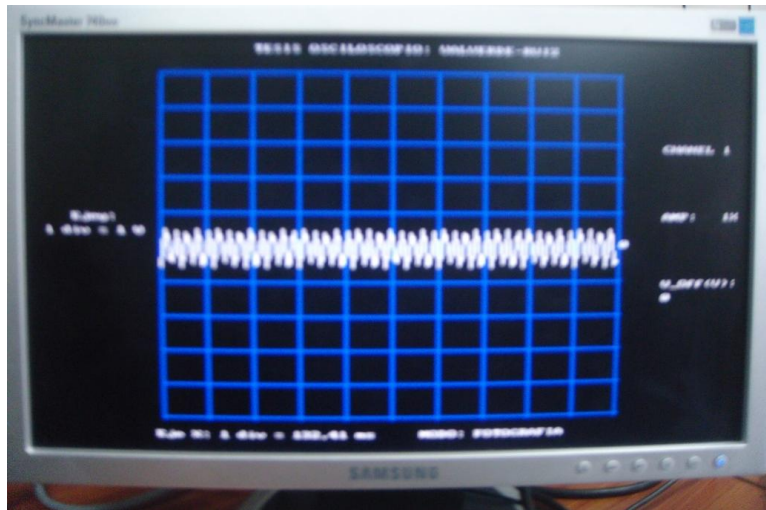
Frecuencia de Señal = 1 Hz, Frecuencia de Muestreo = 6136 Hz



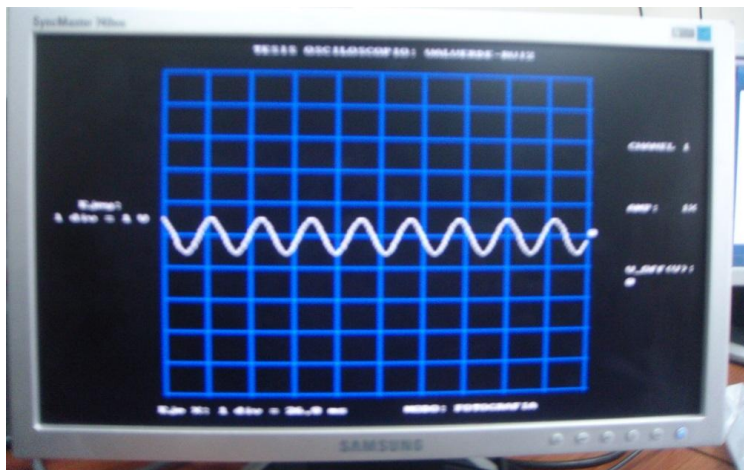
Frecuencia de Señal = 1 Hz, Frecuencia de Muestreo = 7812 Hz



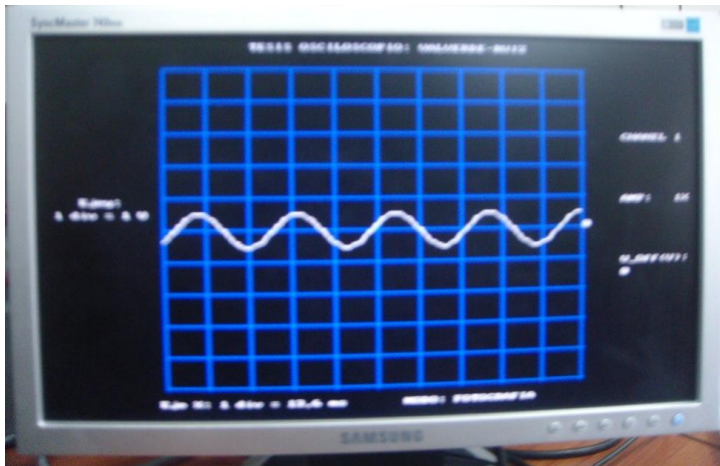
Frecuencia de Señal = 30 Hz, Frecuencia de Muestreo = 138,7 Hz



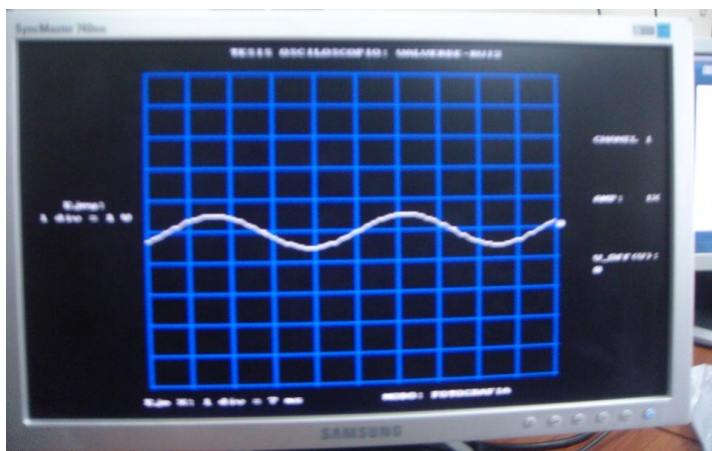
Frecuencia de Señal = 30 Hz, Frecuencia de Muestreo = 693,1 Hz



Frecuencia de Señal = 30 Hz, Frecuencia de Muestreo = 1383 Hz



Frecuencia de Señal = 30 Hz, Frecuencia de Muestreo = 275 Hz



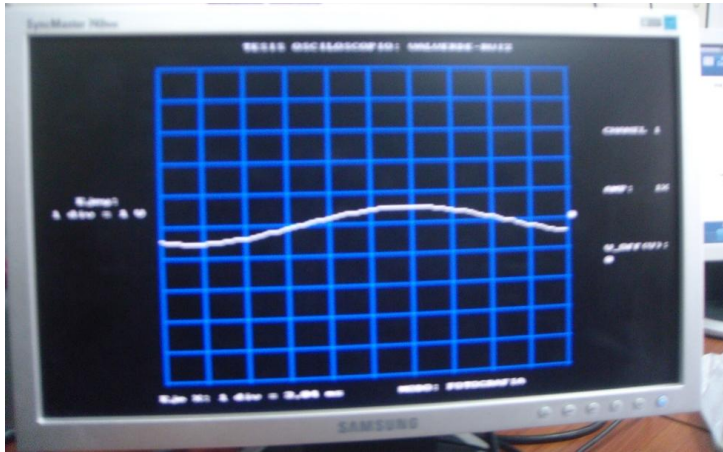
Frecuencia de Señal = 30 Hz, Frecuencia de Muestreo = 4132 Hz



Frecuencia de Señal = 30 Hz, Frecuencia de Muestreo = 4975 Hz



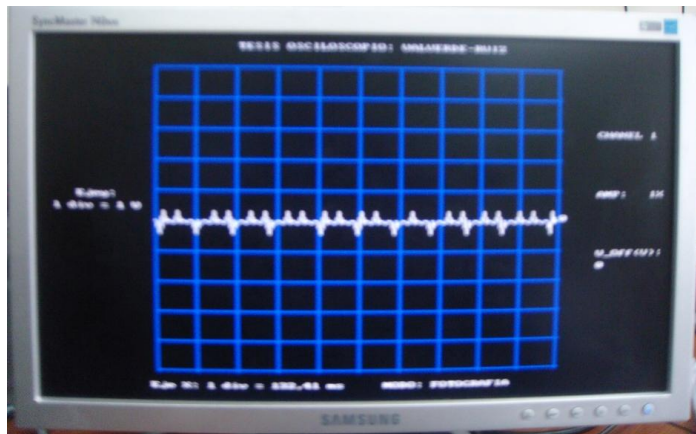
Frecuencia de Señal = 30 Hz, Frecuencia de Muestreo = 6136 Hz



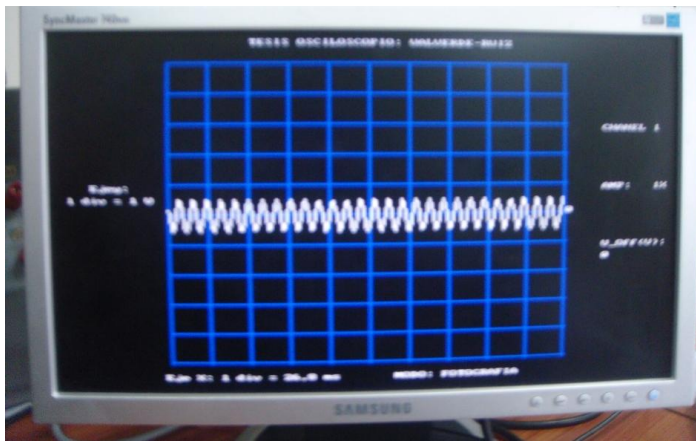
Frecuencia de Señal = 30 Hz, Frecuencia de Muestreo = 7812 Hz



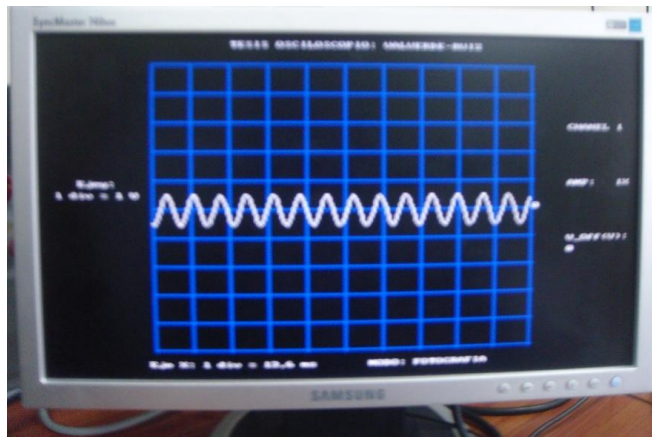
Frecuencia de Señal = 100 Hz, Frecuencia de Muestreo = 138,7 Hz



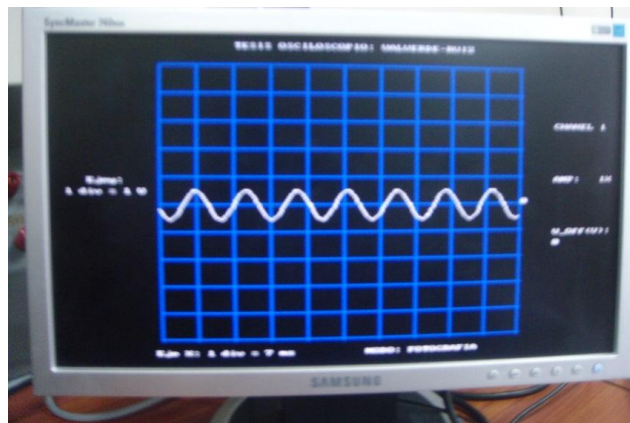
Frecuencia de Señal = 100 Hz, Frecuencia de Muestreo = 693,1 Hz



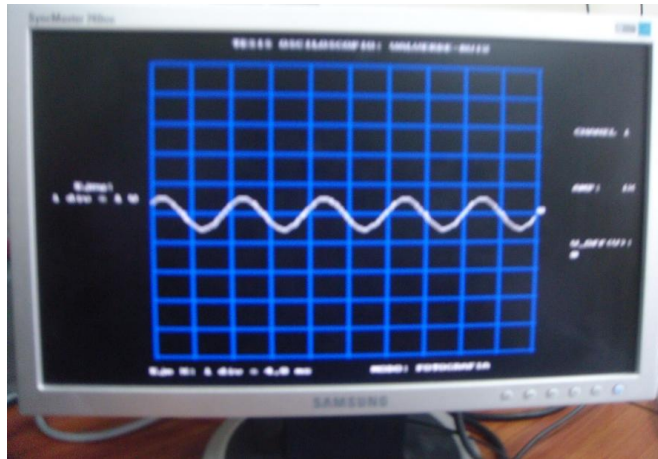
Frecuencia de Señal = 100 Hz, Frecuencia de Muestreo = 1383 Hz



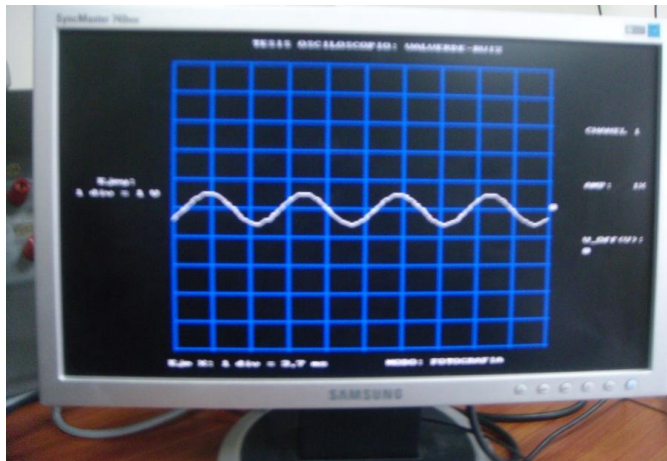
Frecuencia de Señal = 100 Hz, Frecuencia de Muestreo = 2755 Hz



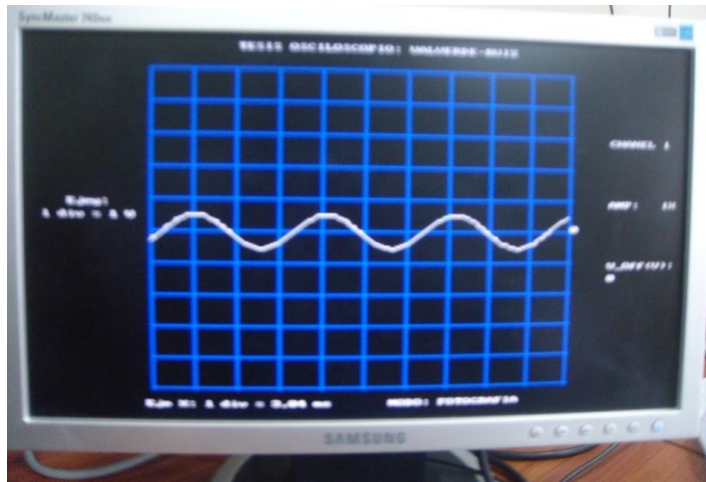
Frecuencia de Señal = 100 Hz, Frecuencia de Muestreo = 4132 Hz



Frecuencia de Señal = 100 Hz, Frecuencia de Muestreo = 4975 Hz



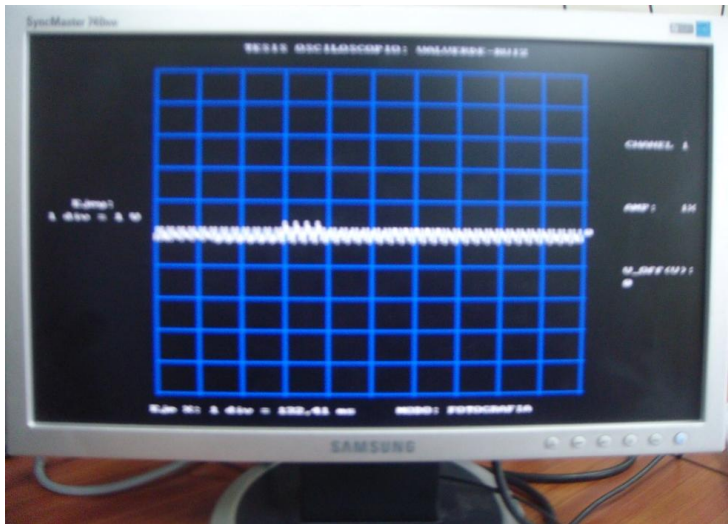
Frecuencia de Señal = 100 Hz, Frecuencia de Muestreo = 6136 Hz



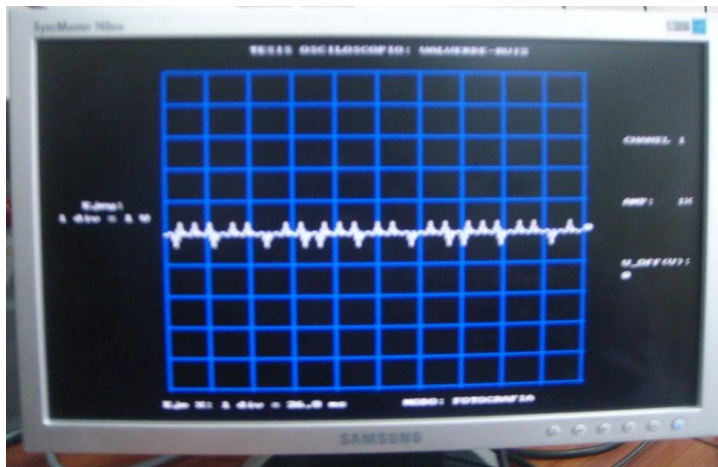
Frecuencia de Señal = 100 Hz, Frecuencia de Muestreo = 7812 Hz



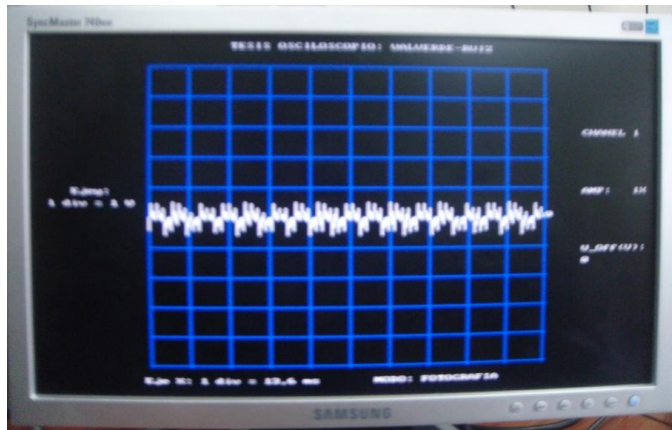
Frecuencia de Señal = 500 Hz, Frecuencia de Muestreo = 138,7 Hz



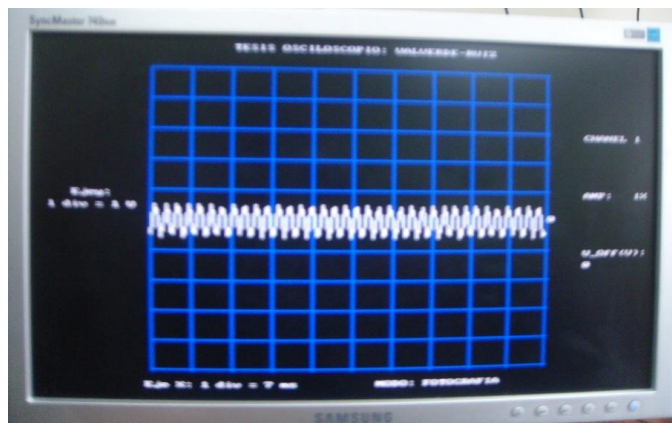
Frecuencia de Señal = 500 Hz, Frecuencia de Muestreo = 693,1 Hz



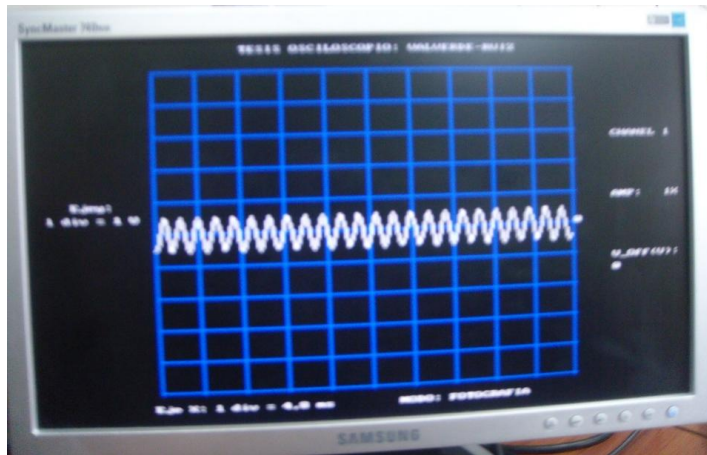
Frecuencia de Señal = 500 Hz, Frecuencia de Muestreo = 1383 Hz



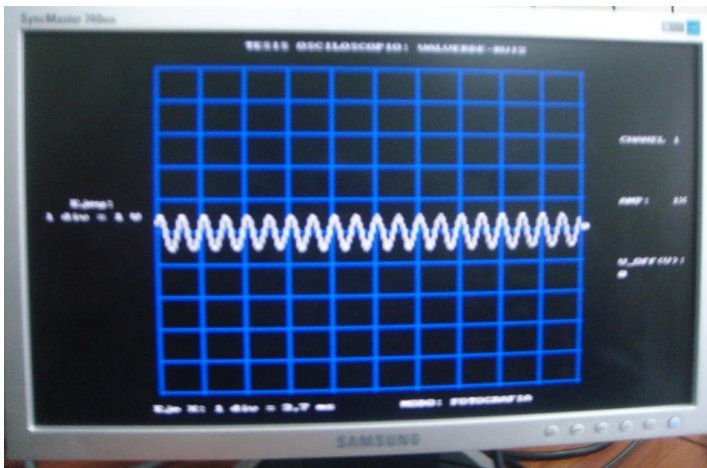
Frecuencia de Señal = 500 Hz, Frecuencia de Muestreo = 2755 Hz



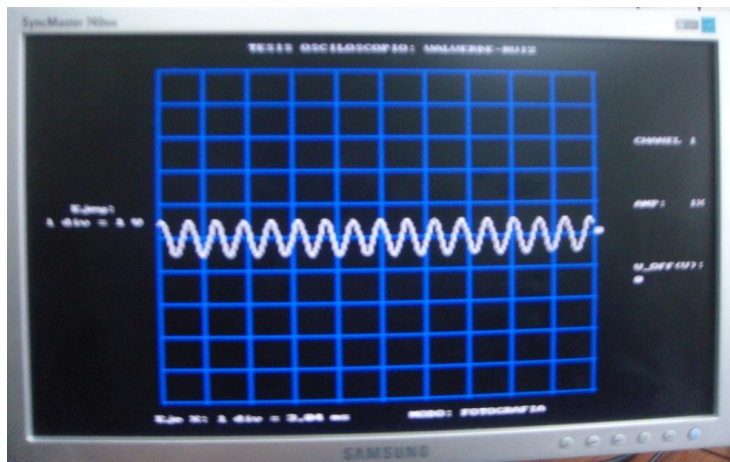
Frecuencia de Señal = 500 Hz, Frecuencia de Muestreo = 4132 Hz



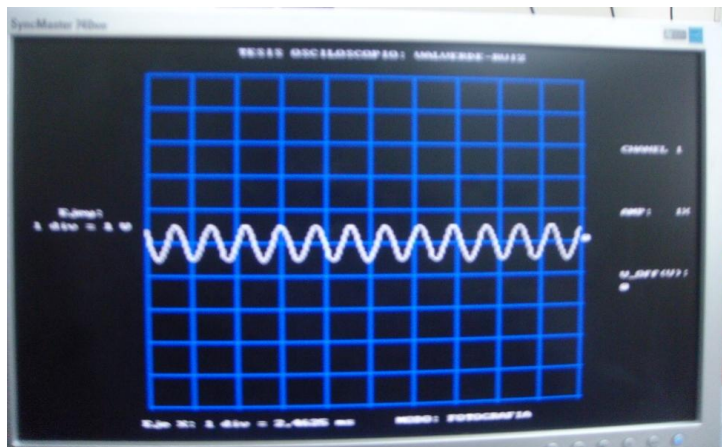
Frecuencia de Señal = 500 Hz, Frecuencia de Muestreo = 4975 Hz



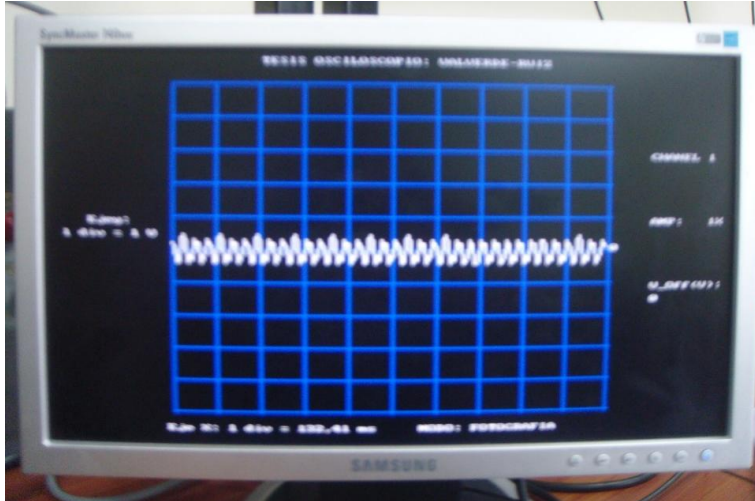
Frecuencia de Señal = 500 Hz, Frecuencia de Muestreo = 6136 Hz



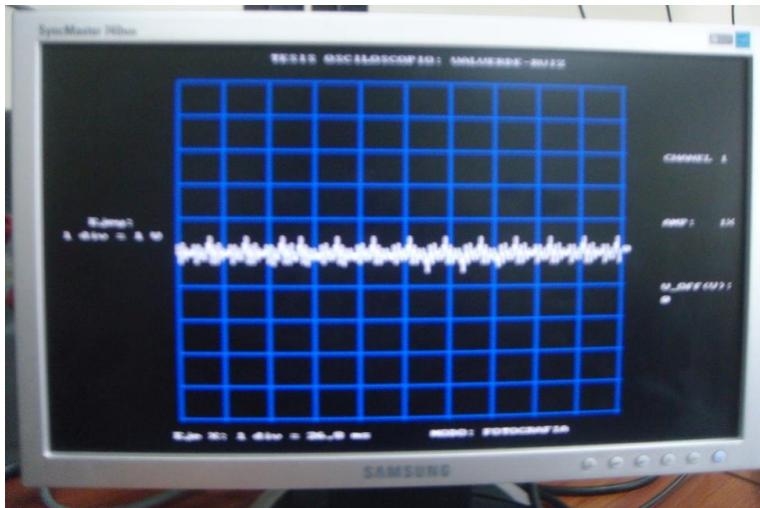
Frecuencia de Señal = 500 Hz, Frecuencia de Muestreo = 7812 Hz



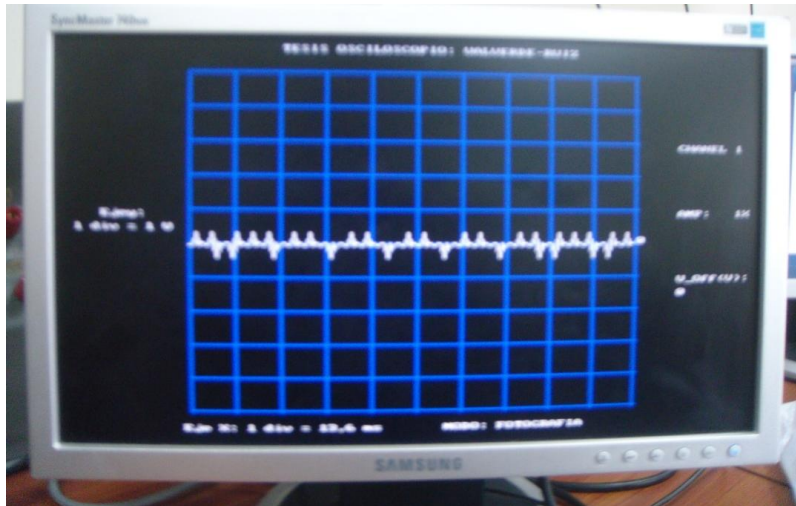
Frecuencia de Señal = 1000 Hz, Frecuencia de Muestreo = 138,7 Hz



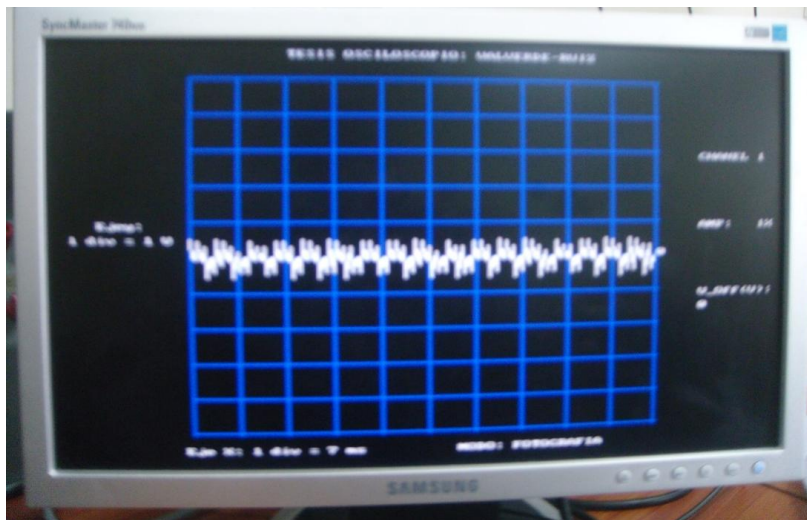
Frecuencia de Señal = 1000 Hz, Frecuencia de Muestreo = 693,1 Hz



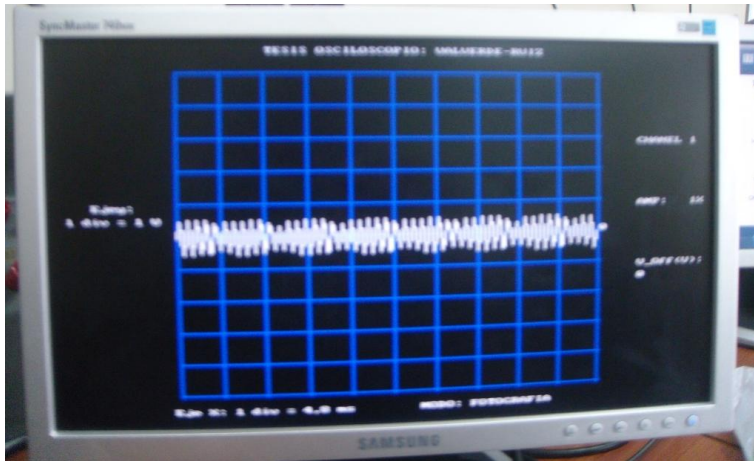
Frecuencia de Señal = 1000 Hz, Frecuencia de Muestreo = 1383 Hz



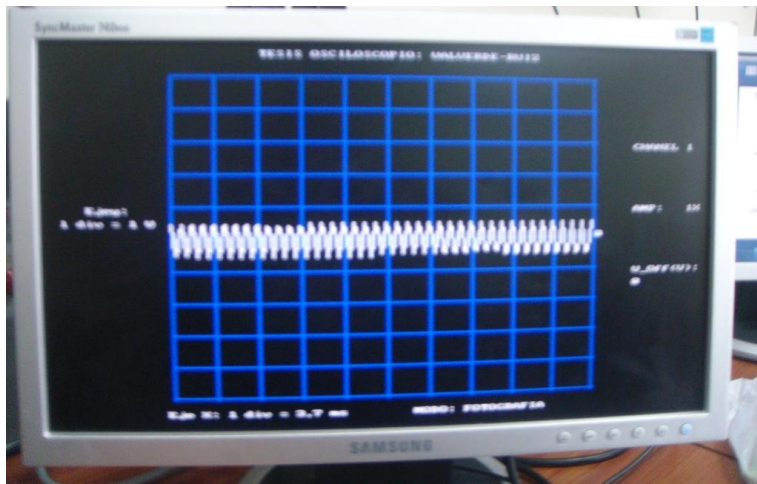
Frecuencia de Señal = 1000 Hz, Frecuencia de Muestreo = 2755 Hz



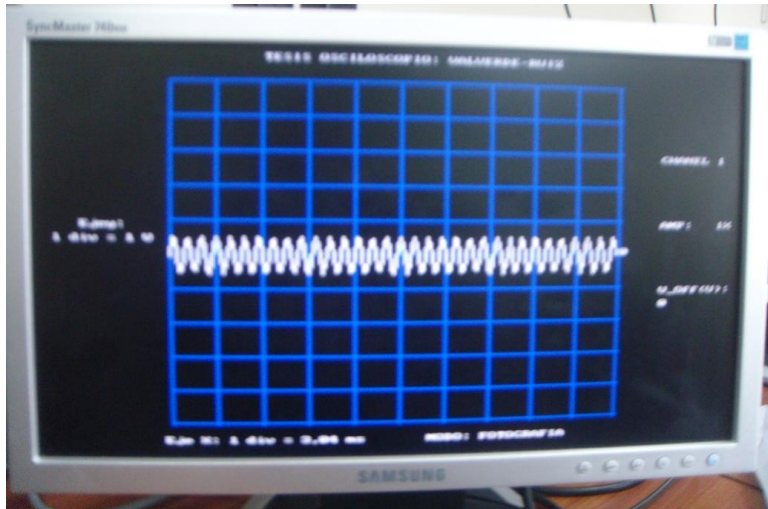
Frecuencia de Señal = 1000 Hz, Frecuencia de Muestreo = 4132 Hz



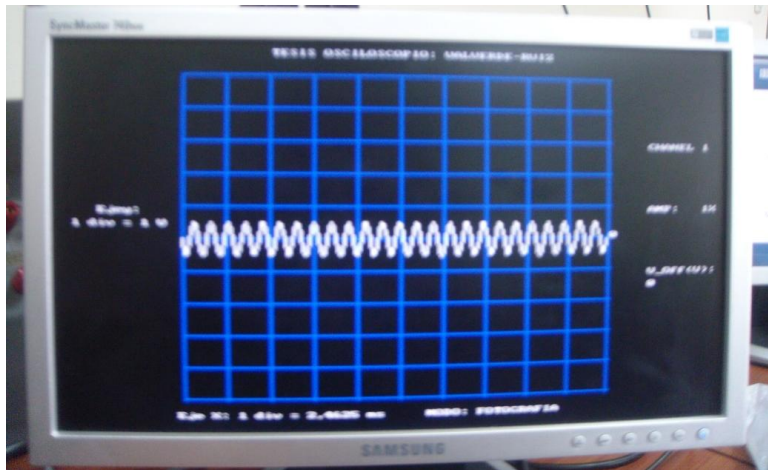
Frecuencia de Señal = 1000 Hz, Frecuencia de Muestreo = 4975 Hz



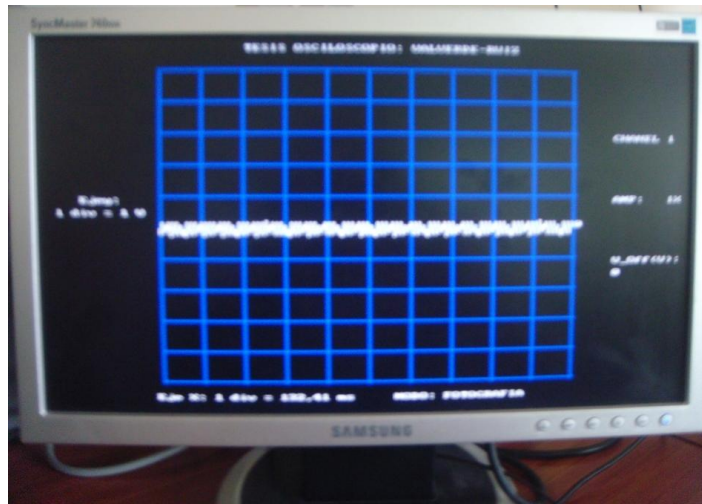
Frecuencia de Señal = 1000 Hz, Frecuencia de Muestreo = 6136 Hz



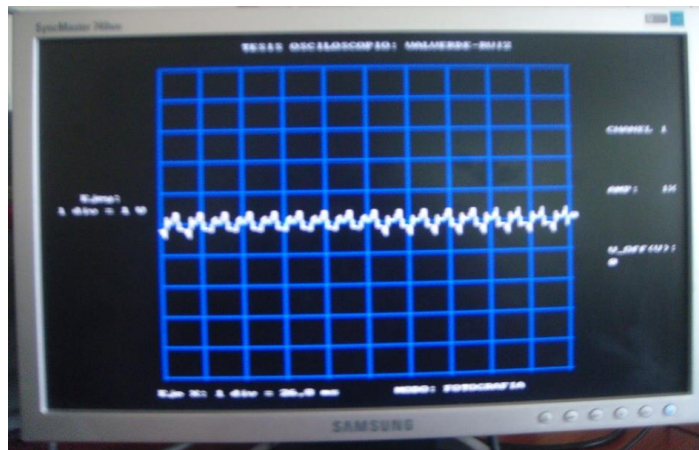
Frecuencia de Señal = 1000 Hz, Frecuencia de Muestreo = 7812 Hz



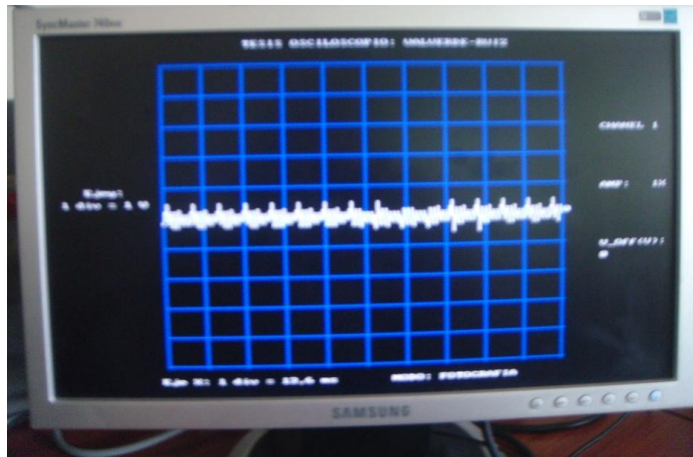
Frecuencia de Señal = 2000 Hz, Frecuencia de Muestreo = 138,7 Hz



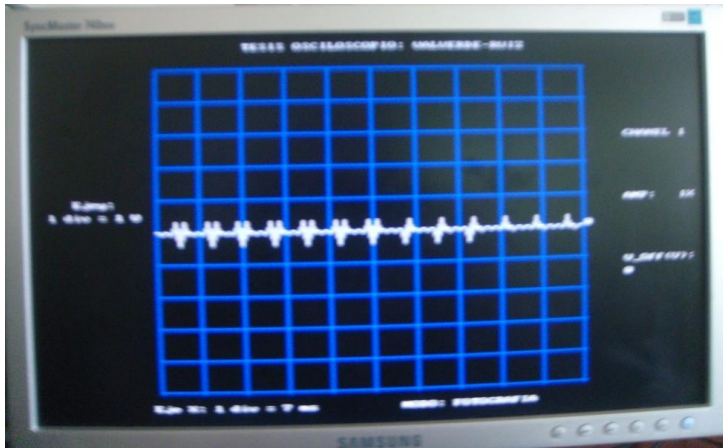
Frecuencia de Señal = 2000 Hz, Frecuencia de Muestreo = 693,1 Hz



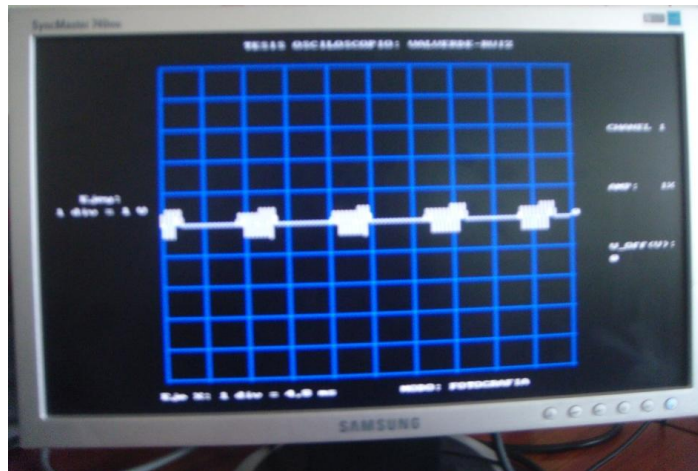
Frecuencia de Señal = 2000 Hz, Frecuencia de Muestreo = 1383 Hz



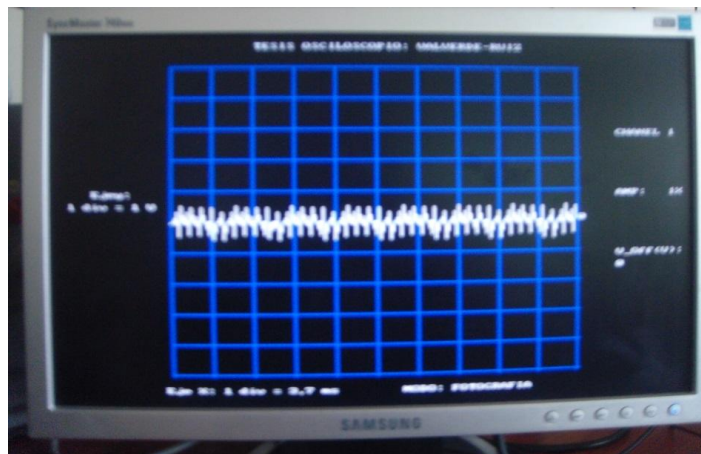
Frecuencia de Señal = 2000 Hz, Frecuencia de Muestreo = 2755 Hz



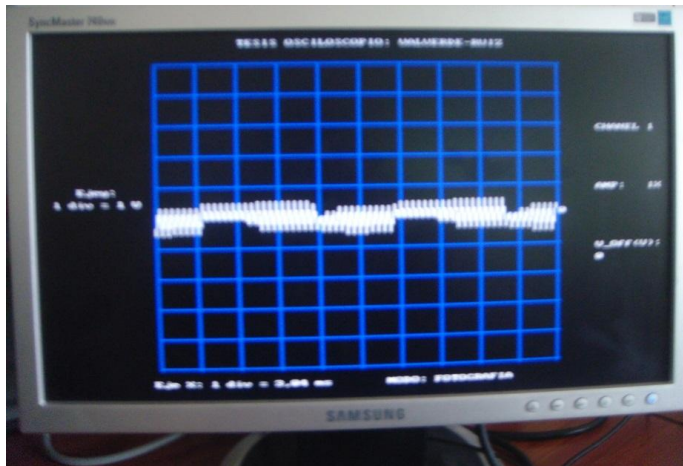
Frecuencia de Señal = 2000 Hz, Frecuencia de Muestreo = 4132 Hz



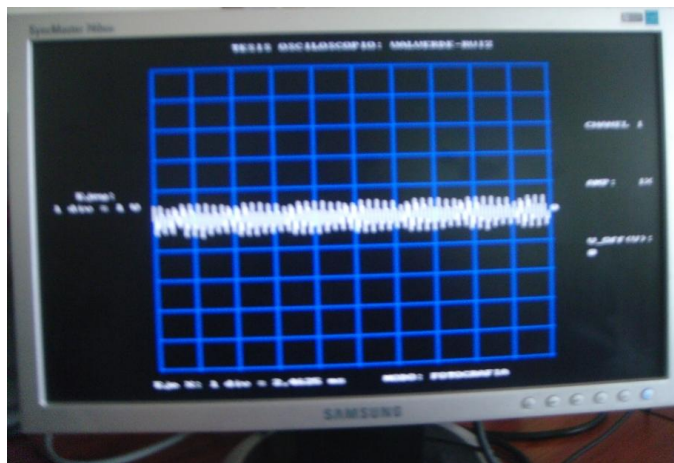
Frecuencia de Señal = 2000 Hz, Frecuencia de Muestreo = 4975 Hz



Frecuencia de Señal = 2000 Hz, Frecuencia de Muestreo = 6136 Hz



Frecuencia de Señal = 2000 Hz, Frecuencia de Muestreo = 7812 Hz



BIBLIOGRAFIA

- [1] Pallás Areny, Ramón. "Instrumentos Electrónicos Básicos". Marcombo Copyright. 2006. Pag 161.
- [2] Reza Langari, Alan S Morris. "Measurement and Instrumentation: Theory and Application". Academic Press. Copyright, Elsevier Inc. 2012. Pag. 173
- [3] Trundle, Eugene. "Servicing Tv, Satellite and Video Equipment". Newnes Copyright. 2002. Pag 15.
- [4] Altera Corporation, FPGA, <http://www.altera.com/products/fpga.html>
- [5] Ronald Sass, Andrew G. Schmidt. "Embedded Systems Design with Platform FPGAs: Principles and Practices". Morgan Kaufmann. Estados Unidos de América, 2010. Pag 21.
- [6] Altera Corporation. "Nios II Processor Reference Handbook". www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf. 3 de Mayo de 2011.
- [7] Altera Corporation. "Media Computer System for the Altera DE2 Board". ftp://ftp.altera.com/up/pub/Altera_Material/11.0/Examples/DE2/NiosII_Computer_Systems/DE2_Media_Computer.pdf. Marzo de 2009.
- [8] Altera Corporation. "Video Out IP Cores for Altera DE Boards".

ftp://ftp.altera.com/up/pub/University_Program_IP_Cores/90/Video_Out.pdf. Marzo de 2009.

- [9] Altera Corporation. "DE2 Development and Education Board User Manual".
ftp://ftp.altera.com/up/pub/Webdocs/DE2_UserManual.pdf. 2006.
- [10] Chu, Pong P. "Embedded SOPC Design with NIOS II Processor and VHDL Examples". A John Wiley & Sons. 2011.
- [11] Yi Yao. "Oscilloscope". <http://yyao.ca/projects/oscilloscope/>. Enero 2006.
- [12] "ADC0801 ADC0802 ADC0803 ADC0804 ADC0805 8-Bit mP Compatible A/D Converters". National Semiconductor. Diciembre 1994.
- [13] Couch, Leon W. "Sistemas de comunicación digitales y analógicos". Quinta Edición. Pearson Educación. Pag 86 – 89.