

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Procesamiento Digital de Señales

Apuntes de clase sobre el Tratamiento Digital de Señales en Tiempo Real

A. Osman

Universidad de Carabobo

9 de diciembre de 2018

Contenido

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

- 1 Introducción
- 2 Técnicas de diseño
- 3 Plataforma de trabajo
- 4 Referencias

El concepto de tratamiento discreto

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

El tratamiento discreto, es la segunda etapa del procesamiento discreto de señales en tiempo continuo y es la encargada de realizar las modificaciones pertinentes a la señal discreta para que produzca el efecto requerido en la señal recuperada. En esta etapa existen dos conceptos teóricos clave para el diseño de sistemas discretos: la respuesta en frecuencia efectiva y la invarianza al impulso.

Diagrama de bloque

Diagrama general del tratamiento discreto de señales en tiempo continuo

PDS

A. Osman

Introducción

Técnicas de diseño

Plataforma de trabajo

Referencias

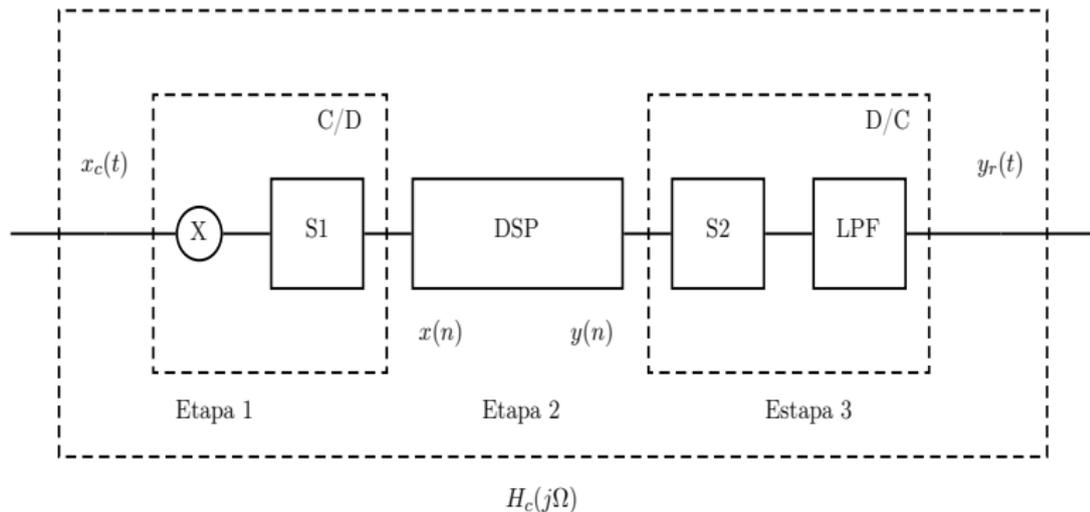


Diagrama de bloque(continuación)

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Del diagrama de bloque se tienen tres etapas

- Etapa1: Muestreo
- Etapa2: Tratamiento
- Etapa3: Reconstrucción

S_1 y S_2 son sistemas de conversión de tren de impulsos a secuencias y viceversa, respectivamente, y sus estructuras internas se estudiaron con mayor profundidad al abordar los conversores A/D y D/A.

Respuesta en frecuencia efectiva

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

En el diagrama de bloque presentado anteriormente $H(j\Omega)$ generalmente, es de espectro ilimitado. Ahora bien, si además se cumple que:

- El sistema DSP es LTI
- $x(t)$ es de banda limitada
- Se muestreó a la rata de Nyquist

entonces el sistema DSP será un sistema equivalente y cumplirá las misma función que un sistema analógico y su diseño se podrá hacer a través de la obtención de una *respuesta en frecuencia efectiva* $H_e(j\Omega)$

$$H_e(j\Omega) = \begin{cases} H(e^{j\Omega T_s}) & \text{si } |\Omega| < \frac{\pi}{T_s} \\ 0 & \text{si } |\Omega| \geq \frac{\pi}{T_s} \end{cases}$$

Relación de diseño

PDS

A. Osman

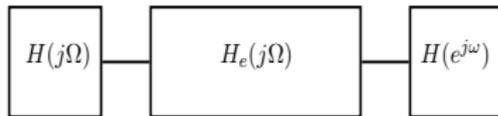
Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

El siguiente diagrama no representa un proceso como tal, es un diagrama que intenta explicar el hecho de que para lograr obtener $H(e^{j\omega})$ a partir de $H(j\Omega)$ o viceversa, es recomendable (y en esto consiste en método de la respuesta en frecuencia efectiva) determinar previamente $H_e(j\Omega)$ [1]



sustituir $\Omega = \frac{\omega}{T_s}$

aperiódico e infinito

aperiódico y finito

periódico cada 2π

Ejemplo

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Para el sistema diferenciador ideal para señales en tiempo continuo de banda limitada, definido como:

$$y(t) = \frac{d}{dt}[x(t)]$$

Determinar:

$\Rightarrow H(j\Omega)$ y sus gráficos de magnitud y fase.

$\Rightarrow H_e(j\Omega)$ y sus gráficos de magnitud y fase.

$\Rightarrow H(e^{j\omega})$ y sus gráficos de magnitud y fase.

$\Rightarrow h[n]$.

Ejemplo (continuación)

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

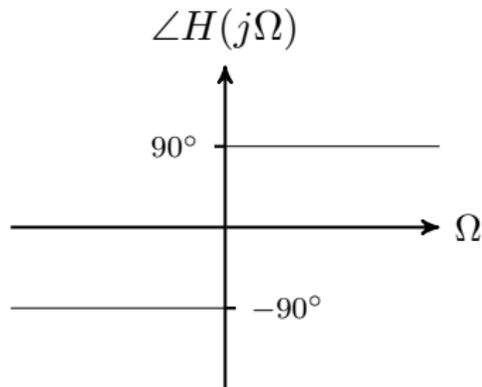
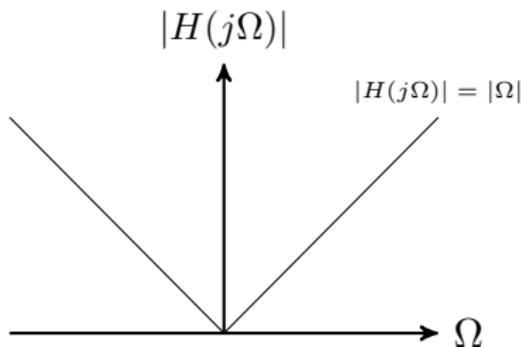
Referencias

Aplicando la transformada de Fourier en tiempo continuo

$$y(t) = \frac{d}{dt}[x(t)] \longleftrightarrow Y(j\Omega) = j\Omega X(j\Omega)$$

En un sistema LTI, $Y(j\Omega) = H(j\Omega) X(j\Omega)$, por tanto

$$H(j\Omega) = j\Omega$$



Ejemplo (continuación)

PDS

A. Osman

Introducción

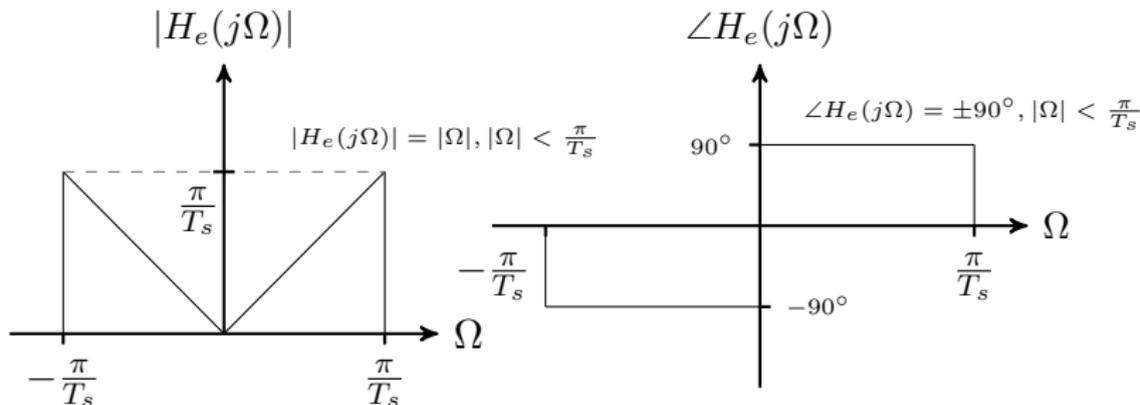
Técnicas de
diseño

Plataforma
de trabajo

Referencias

Dado que las señales de tiempo continuo son de banda limitada, la definición de $H_e(j\Omega)$ es la siguiente:

$$H_e(j\Omega) = \begin{cases} H(j\Omega) & \text{si } |\Omega| < \frac{\pi}{T_s} \\ 0 & \text{si } |\Omega| \geq \frac{\pi}{T_s} \end{cases}$$



Ejemplo(continuación)

PDS

A. Osman

Introducción

Técnicas de
diseño

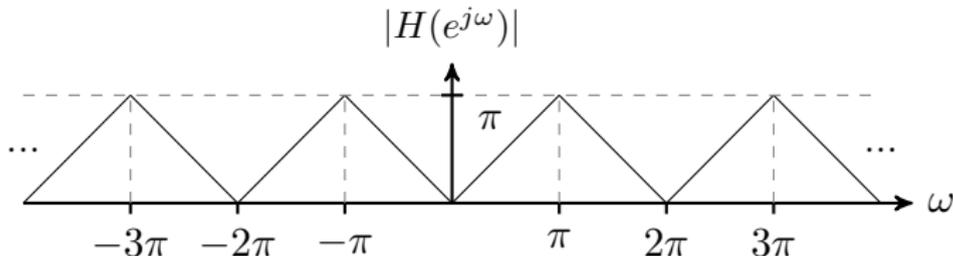
Plataforma
de trabajo

Referencias

$H(e^{j\omega})$ la obtenemos a partir de $H_e(j\Omega)$, utilizando el factor de escalamiento $\omega = \Omega T_s$, es decir,

$$H(e^{j\omega}) = H_e(j\Omega) \big|_{\Omega=\omega/T_s}$$

Siendo $H(e^{j\omega})$ periódica con periodo 2π



Ejemplo (continuación)

PDS

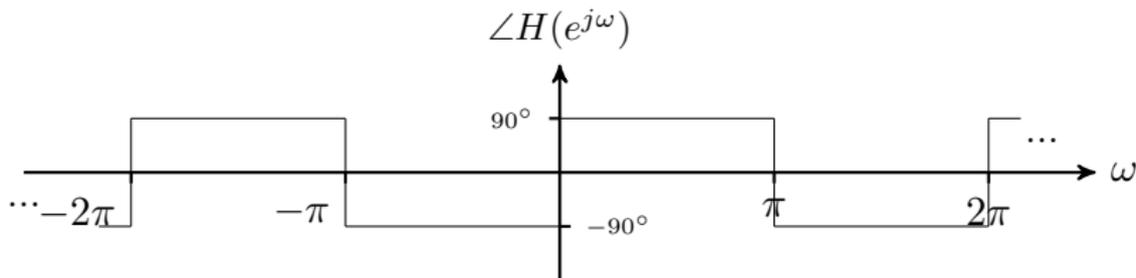
A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias



Para hallar $h[n]$ aplicaremos la definición de la transformada inversa:

$$h[n] = \frac{1}{2\pi} \int_{2\pi} H(e^{j\omega}) e^{j\omega n} d\omega$$

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{j\omega}{T_s} e^{j\omega n} d\omega = \frac{j}{2\pi T_s} \int_{-\pi}^{\pi} \omega e^{j\omega n} d\omega$$

Ejemplo (continuación)

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Resolviendo la integral resulta:

$$h[n] = \frac{n\pi \cos(\pi n) - \sin(\pi n)}{n^2\pi T_s}$$

Como $\sin(\pi n) = 0$, para todo n :

$$h[n] = \frac{\cos(\pi n)}{nT_s}$$

Se puede demostrar que para éste caso $h[0] = 0$

Invarianza al impulso

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

En el caso específico en que $H_e(j\Omega) = H(j\Omega)$, esto es que $H(j\Omega)$ sea de banda limitada y que $X(j\Omega)$ también lo sea, entonces el sistema discreto se puede determinar a partir de la técnica denominada *invarianza al impulso*[1], que consiste en aplicar las siguientes expresiones:

$$H(e^{j\omega}) = H\left(j\frac{\omega}{T_s}\right)$$

$$h[n] = T_s h(nT_s)$$

Ecuación general de los sistemas en tiempo discreto

PDS

A. Osman

Introducción

Técnicas de diseño

Plataforma de trabajo

Referencias

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \quad (1)$$

o también

$$y[n] - \sum_{k=1}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \quad (2)$$

Función de transferencia de los sistemas en tiempo discreto

PDS

A. Osman

Introducción

Técnicas de diseño

Plataforma de trabajo

Referencias

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}} \quad (3)$$

Flujograma de un sistema en tiempo discreto

Forma Direta I

PDS

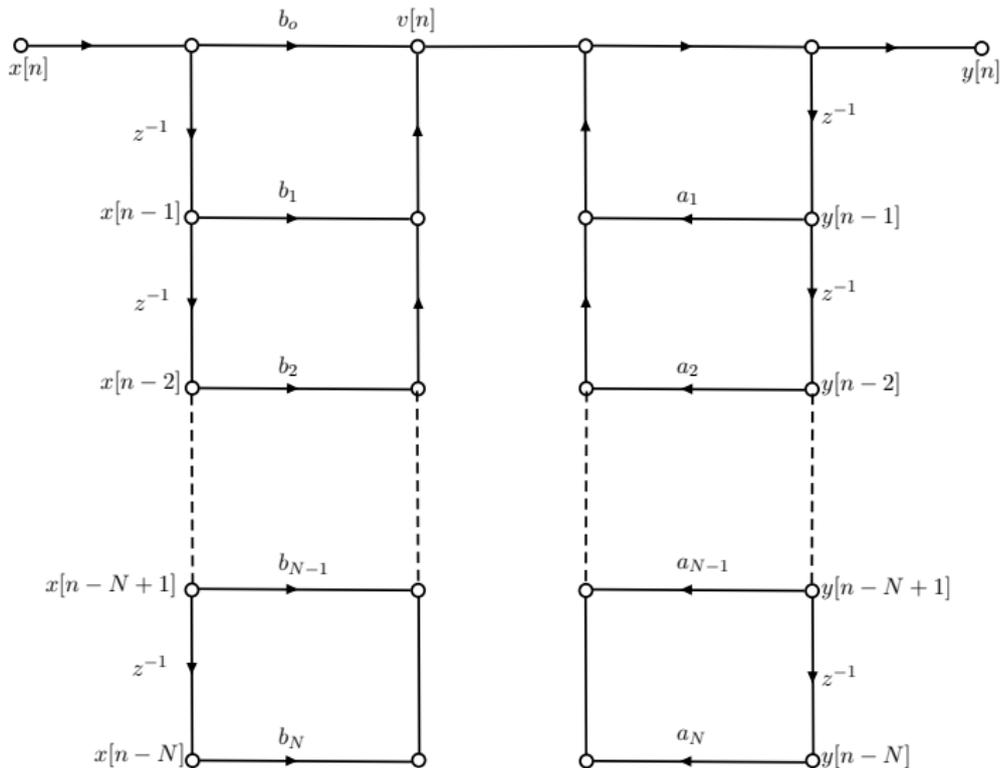
A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias



Flujograma de un sistema en tiempo discreto

Forma Directa II

PDS

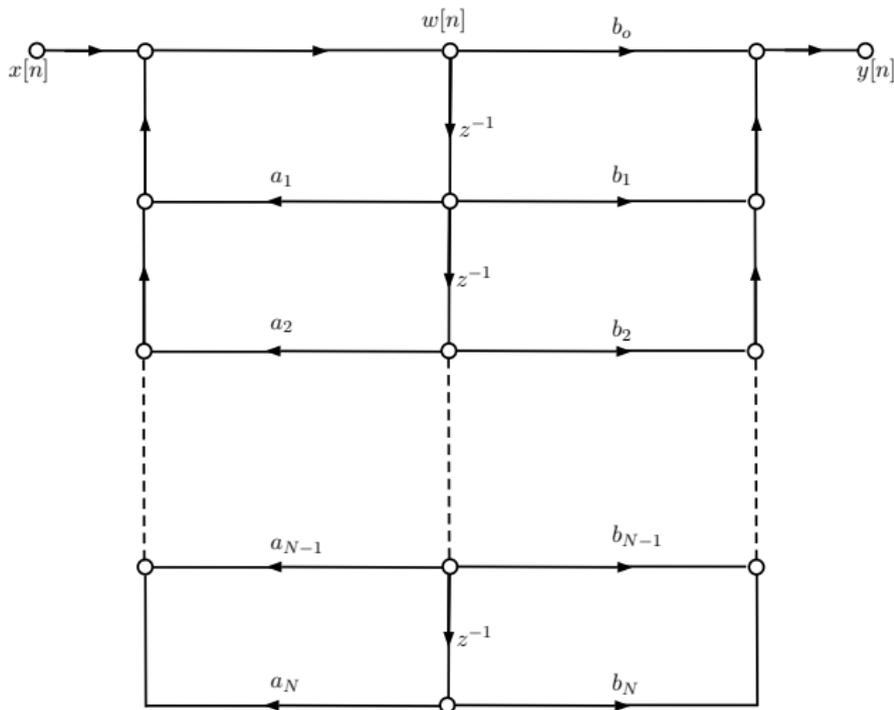
A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias



Plataforma de trabajo

El dsk tms320c55x

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Una de las opciones para implementar el tratamiento de señales digitales en tiempo real es mediante el uso de Hardware especializado, en el cual es necesaria la programación de algoritmos sobre placas de desarrollo. Un ejemplo es la tarjeta Texas Instruments de la serie tms320c55x cuya programación se realiza principalmente en lenguaje C a través del entorno de desarrollo integrado Code Composer Studio.

Plataforma de trabajo

Diagrama de funcionamiento

PDS

A. Osman

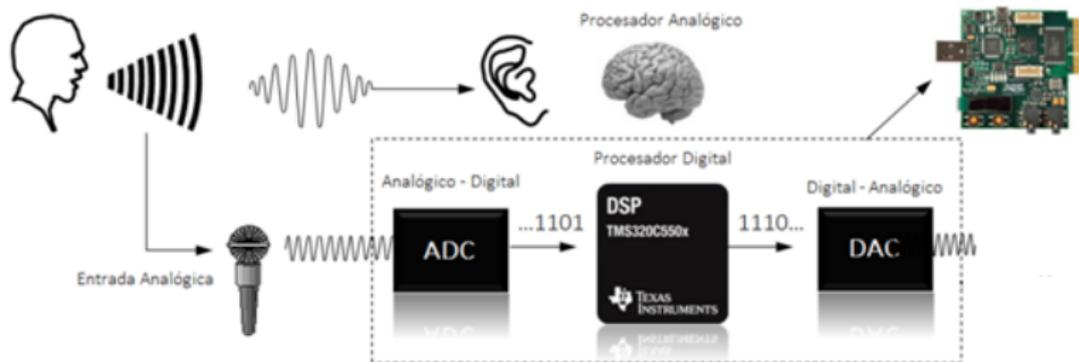
Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Figura 1: Diagrama de procesamiento en tiempo real. Fuente: [2]



Plataforma de trabajo

Esquema de conexión

PDS

A. Osman

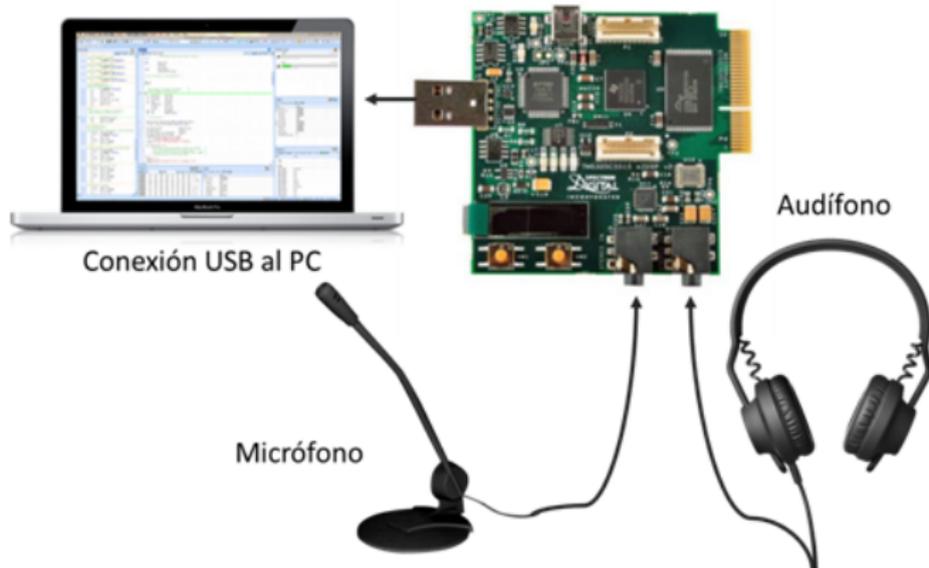
Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Figura 2: Esquema de conexión. Fuente: [2]



Plataforma de trabajo

Esquema de conexión

PDS

A. Osman

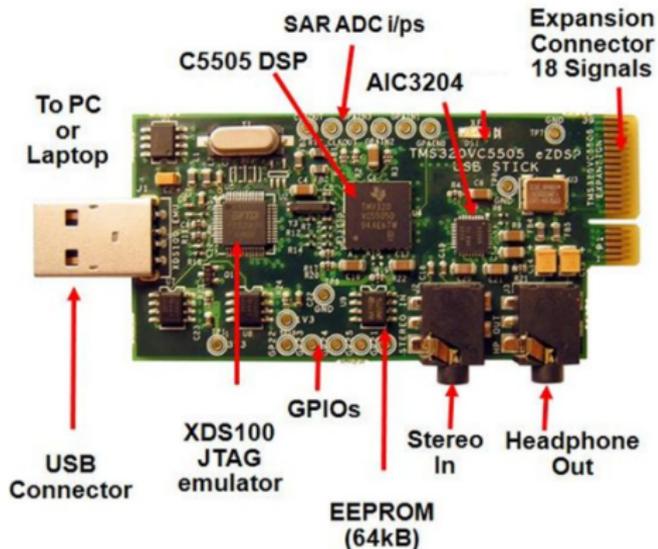
Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Figura 3: Elementos. Fuente: [2]



Plataforma de trabajo

Siglas de la Tarjeta de Desarrollo

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

- **DSK:** El procesador de señales viene integrado a una tarjeta, junto a una serie de componentes, por ejemplo: codificador de audio, memorias, periféricos, etc.
- **TMS:** Dispositivo calificado, al ser elaborado por la Texas Instruments.
- **C:** Tecnología de construcción CMOS (tipo de Transistores utilizados en la construcción).
- **5:** Generación de la tarjeta.
- **x:** Versión de la tarjeta (05, 15, 35, etc).

Plataforma de trabajo

Puertos de E/S y Conexiones

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

- **USB Connector:** Permite la entrada y salida de datos a través de un computador o laptop. Además, ofrece alimentación a la tarjeta.
- **C5505 DSP:** DSP encargado de realizar el procesamiento digital de la señal que es enviada desde los puertos de entrada, el cual es de punto fijo, esto quiere decir que trabaja con valores enteros. Capaz de ejecutar 200 millones de instrucciones por segundo.
- **SAR ADC i/ps:** Puertos de entrada/salida al convertidor analógico/digital
- **AIC3240:** Codifica y decodifica señales de audio, a una frecuencia de muestreo de 48kHz

Plataforma de trabajo

Puertos de E/S y Conexiones

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

- **LED:** Bombillo LED que permite mostrar salidas simples (encendido o apagado).
- **Expansion Connector 18 Signals:** Utilizado para llevar las señales de salida procesadas por el DSP a una interfaz de usuario.
- **Headphone Out:** Puerto a través del cual se recibe la señal procesada, de audio en nuestro caso.

Plataforma de trabajo

Puertos de E/S y Conexiones

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

- **Stereo In:** Puerto a través del cual se puede enviar una señal de audio para ser procesada en la tarjeta DSP.
- **EEPROM:** Memoria en la que se almacena el código que ejecuta la tarjeta y las muestras adquiridas.
- **GPIOs:** Puertos de entrada/salida de propósito general. Pueden usarse en pantallas LCD, alarmas, interfaces de comunicación, entre otros.
- **XDS100 JTAG emulator:** Permite interactuar con la conexión al JTAG de la tarjeta a través del puerto USB. Útil además como mecanismo de depuración de aplicaciones.

Plataforma de trabajo

Code Composer Studio

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Code Composer Studio es un entorno de desarrollo integrado (IDE) que admite el portafolio de microcontroladores y procesadores integrados de TI. También comprende un conjunto de herramientas utilizadas para desarrollar y depurar aplicaciones integradas. Incluye un compilador optimizador C / C ++, editor de código fuente, entorno de compilación de proyecto, depurador, generador de perfiles y muchas otras características.

Plataforma de trabajo

Code Composer Studio

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

El IDE intuitivo proporciona una interfaz de usuario única que lo guía a través de cada paso del flujo de desarrollo de la aplicación. Las herramientas e interfaces familiares permiten a los usuarios comenzar más rápido que nunca. Code Composer Studio combina las ventajas del marco del software Eclipse con capacidades de depuración embebidas avanzadas de TI, lo que resulta en un entorno de desarrollo rico en funciones para desarrolladores integrados.

Plataforma de trabajo

Lenguaje C

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

C es un lenguaje de medio nivel pero con muchas características de bajo nivel, como construcciones del lenguaje que permiten un control, pero también dispone de las estructuras típicas de los lenguajes de alto nivel. Cabe señalar que los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

Plataforma de trabajo

Lenguaje C

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

El compilador C traduce programas de alto nivel a lenguaje de máquina que puede ser ejecutado por procesadores como los pertenecientes a la serie TMS320C55x. Por otro lado, los ambientes de programación incluyen programas de depuración (debugger) para identificar errores en programas fuente.

Plataforma de trabajo

DSPLib

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

La biblioteca DSP LIB contiene un paquete de procedimientos útiles para el uso en el área de procesamiento digital de señales, estas rutinas ya están programadas e implementadas para su uso en las tarjetas texas instruments, las rutinas incluidas dentro de la biblioteca están organizadas en ocho diferentes categorías funcionales.

Plataforma de trabajo

DSPLib/Categorías funcionales

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

- Transformadas de Fourier rápida (FFT)
- Filtrado y convolución
- Filtrado adaptativo
- Correlación
- Matemático
- Trigonométrico
- Diverso
- Matriz

Plataforma de trabajo

DSPLib/Funciones significativas

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

Algunas funciones muy importantes son las siguientes:

- Cfft (Forward Complex FFT): que se usa de esta forma 'void cfft (DATA *x, ushort nx, type)', sirve para calcular una FFT compleja de nx en el vector x.
- Cfir(Complex FIR Filter): se usa de esta forma 'ushort oflag = cfir (DATA *x, DATA *h, DATA *r, DATA *dbuffer, ushort nx, ushort nh)', sirve para calcular un filtro FIR complejo (forma directa) utilizando los coeficientes almacenados en vector h.

Plataforma de trabajo

DSPLib/Funciones significativas(continuación)

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

- Convolution: se usa de esta forma `ushort oflag = convol (DATA *x, DATA *h, DATA *r, ushort nr, ushort nh)`, sirve para calcular la convolución real de dos vectores reales `x` y `h`, y coloca el resultados en el vector `r`.
- Y muchas otras funciones que facilitan inmensamente el trabajo de programadores y desarrolladores al momento de interactuar con los dispositivos anteriormente mencionados.

Estructura C

Plantilla de trabajo

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias

```
#include "usbstk5505.h"
#include "aic3204.h"
#include "PLL.h"
#include "stereo.h"

#define Fs 48000 // Frecuencia de muestreo,
se puede utilizar 48kHz,
// 24kHz, 16kHz, 12kHz, 9600Hz, 8kHz y 6857Hz.
#define G 0 // Ganancia (dB).
// Si se utiliza micrófono utilizar 30dB.
// Declaración de variables
void main(void)
{
  USBSTK5505_init( ); // Inicialización
  //de la tarjeta
  aic3204_hardware_init( ); // Configuración del
  //codec en la tarjeta
  aic3204_init( ); // Inicialización del codec
  //de audio AIC3204
  set_sampling_frequency_and_gain(Fs, G);
  // Conf. de frecuencia de muestreo y ganancia
  // Coloque código aquí
  aic3204_disable( ); // Desabilitar codec
  //AIC3204
  SW_BREAKPOINT;
}
```

Referencias

PDS

A. Osman

Introducción

Técnicas de
diseño

Plataforma
de trabajo

Referencias



Ronald W. Oppenheim, Alan V.; Schafer.

Tratamiento de Señales en tiempo discreto.

Prentice Hall. Madrid. ES. 2a ed. 873 p., Reading, MA,
2000.



M. De oliveira.

*Desarrollo de una plataforma e-learning mediante el dsk
tms320c55xx como herramienta base para el laboratorio
de procesamiento digital de señales en tiempo real.*

Escuela de Ingeniería de Teelcomunicaciones UC, NA,
2015.