

VII Ejemplo de Control de instrumentación mediante GPIB.

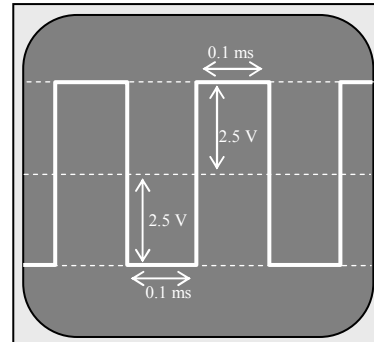
Descripción del equipamiento que se utiliza:

- Se conecta la salida del generador al canal #1 del osciloscopio.
- Se establece el osciloscopio para que responda al GPIB con código 5
- Se establece el generador para que responda al GPIB con código 10.

Desarrollar un programa en el ordenador que realice secuencialmente las siguientes operaciones.

a) Programar el generador de funciones para que genere una señal cuadrada de amplitud 5 Vpp, nivel de continua nulo y 5KHz de frecuencia.

b) Programar el osciloscopio para que visualice la señal generada en pantalla, de forma que la resolución temporal y de amplitud sean óptimas.



c) Requerir del osciloscopio que proporcione el valor de continua y el valor pico a pico de la señal.

d) Transferir la forma de onda del osciloscopio al ordenador, y determinar los primeros 10 armónicos, visualizándolos en un diagrama del ordenador y obteniendo el coeficiente THD de distorsión armónica total de la señal.

$$v_0 = \sum_{i=0}^{\infty} c_i \cos(10^4 \pi t + \phi_n) \quad THD = 100 \sqrt{\sum_{i=2}^{10} \left[\frac{c_i}{c_1} \right]^2}$$

Programa Matlab.

```
% =====
% Ejemplo de programacion de un entorno de instrumentacion.
% Programador: Jose M. Drake
% Fecha: 9-11-03
% =====

% Se inicializa el entorno Matlab.
clear all;
% =====

% Se crea un objeto gpib para controlar el generador
% El GPIB-Address del generador es 10.
generador=visa('agilent','GPIB0::10::0::INSTR');
generador.Timeout=2.0;
generador.EOIMode='on';
```

```

generador.EOSMode='none';

% Se establece la conexión con el equipo.

fopen(generador);

% Se establece el generador para que genere la señal establecida en el
% ejemplo:
% Se inicializa el Parser y el equipo
clrdevice(generador);
fprintf(generador,'*CLS');
fprintf(generador,'*RST');

% Se establece que la carga va ser de alta impedancia.
fprintf(generador,'OUTPUT:Load INFINITY');

% Se programa la señal: cuadrada de 5 Khz y 5 Vpp y offset nulo.
fprintf(generador,'APPLY:Square 5 khz,5 vpp');

% Se establece que la señal sea cuadrada.
fprintf(generador,'PULSE:Dcycle 50');

% =====
% Se crea un objeto gpib para controlar el osciloscopio
% El GPIB-Address del generador es 5.
oscilo=visa('agilent','GPIB0::5::0::INSTR');
oscilo.Timeout=2.0;
oscilo.EOIMode='on';
oscilo.EOSMode='none'
oscilo.inputBufferSize=4256;

% Se establece la conexión con el osciloscopio, se inicializa el parser y
% el equipo, y se inhiben los botones del panel frontal.
fopen(oscilo);
clrdevice(oscilo);
fprintf(oscilo,'*CLS');
fprintf(oscilo,'*RST');
fprintf(oscilo,'System:LOCK on');

% Se programa el osciloscopio para que visualice correctamente la señal
% generada.
fprintf(oscilo,'CHANNEL1:Coupling DC');
fprintf(oscilo,'CHANNEL1:Offset 0.0');
fprintf(oscilo,'CHANNEL1:Probe x10');
fprintf(oscilo,'CHANNEL1:Range 10.0');
fprintf(oscilo,'VIEW Channel1');
fprintf(oscilo,'TIMEBASE:Range 0.7 ms');

fprintf(oscilo,'BLANK CHANNEL2');

```

```

fprintf(oscilo,'TRIGGER:Mode AUTL');
fprintf(oscilo,'TRIGGER:Source EXTERNAL');

% Se programa las características de codificación de la forma de onda.
fprintf(oscilo,'WAVEFORM:Format BYTE');
fprintf(oscilo,'WAVEFORM:Source CHANNEL1');
fprintf(oscilo,'WAVEFORM:Points 2000');
fprintf(oscilo,'ACQUIRE:Type NORMAL');
fprintf(oscilo,'ACQUIRE:Complete 100');

% =====
% Se espera a que el sistema quede programado.
fprintf(oscilo,'Digitize CHANNEL1');

%=====
% Compruebe que la generación de la señal y su visualización son correctas.
input('Pulse Return para continuar...');

% =====
% COMIENZA LA MEDIDA.
% =====
% Requiere del osciloscopio el valor de continua del canal #1
fprintf(oscilo,'MEASURE:SOURCE CHANNEL1');
fprintf(oscilo,'MEASURE:VAVERAGE');
fprintf(oscilo,'MEASURE:VAVERAGE?');

%Leemos el dato y lo mostramos por pantalla
vavg=fscanf(oscilo);
display(strcat('Tension media= ',vavg,' V.));

% =====
% Requiere del osciloscopio el valor pico a pico del canal #1
fprintf(oscilo,'MEASURE:VPP');
fprintf(oscilo,'MEASURE:VPP?');

%Leemos el dato y lo mostramos por pantalla
vpp=fscanf(oscilo);
display(strcat('Tension pico a pico= ',vpp,' V.));

% =====
% Requiere del osciloscopio la forma de onda
fprintf(oscilo,'DIGITIZE Channel1');
fprintf(oscilo,'WAVEFORM:DATA?');

%Leemos los datos
s=binblockread(oscilo);
fprintf(oscilo,'WAVEFORM:XINCREMENT?');
xincrement=str2num(fscanf(oscilo));
fprintf(oscilo,'WAVEFORM:Xorigin?');
xorigin=str2num(fscanf(oscilo));

```

```

fprintf(oscilo,'WAVEFORM:Xreference?');
xreference=str2num(fscanf(oscilo));
fprintf(oscilo,'WAVEFORM:Yincrement?');
yincrement=str2num(fscanf(oscilo));
fprintf(oscilo,'WAVEFORM:Yorigin?');
yorigin=str2num(fscanf(oscilo));
fprintf(oscilo,'WAVEFORM:Yreference?');
yreference=str2num(fscanf(oscilo));

% =====
% Se convierten los valores a adquiridos a valores físicos y se visualizan
v=(s-yreference)*(yincrement)+yorigin;
t=((1:2000)-xreference)*xincrement+xorigin;
plot(t,v);

% =====
% Se calculan los 10 primeros componentes armónicos.
n=1:floor(2000*0.6/0.7);
T=floor(2000*0.2/0.7);
for arm=1:10
a(arm)=2*sum(v(n)'.*cos(2*pi/T*arm*n))/length(n);
b(arm)=2*sum(v(n)'.*sin(2*pi/T*arm*n))/length(n);
c(arm)=sqrt(a(arm).*a(arm)+b(arm).*b(arm));
end;

% =====
% Se visualizan los 10 armónicos en un diagrama de barras.
figure;
stem(1:10,c);

% Calcula el coeficiente de distorsion armonica THD
thd=100*sqrt(sum(c(2:10).*c(2:10))/(c(1)*c(1)));
disp(strcat('THD= ',num2str(thd),' %'));

% =====
% Se cierran los dispositivos y se eliminan los objetos de control.
fclose(generator);
delete(generator);
clear generator;
fclose(oscilo);
delete(oscilo);
clear oscilo;
%=====

```