

Universidad Central de Venezuela  
Facultad de Ingeniería  
Escuela de Ingeniería Eléctrica  
Departamento de Electronica, computación y control  
Microprocesadores I

## **Proyecto Final: Convertidor Digital-Analogico**

**Docente:**

Gutierrez, Ivan

**Estudiante:**

Br. Lopez, Jose

V26893008

## Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Explicación del Proyecto</b>	<b>4</b>
<b>3</b>	<b>Cálculos para generar las bases de tiempo</b>	<b>4</b>
<b>4</b>	<b>Valores usados para la configuración del DAC</b>	<b>4</b>
<b>5</b>	<b>Diagramas de flujo</b>	<b>4</b>
5.1	Main . . . . .	4
5.2	Interrupcion . . . . .	6
5.3	Comunicacion . . . . .	9
<b>6</b>	<b>Conclusiones</b>	<b>10</b>

# 1 Introducción

El tema de este proyecto es el desarrollo de un generador de ondas senoidales utilizando el microcontrolador PIC18F45K50. Este dispositivo estará compuesto por un teclado, una pantalla y un convertidor digital-analógico (DAC) para la producción de señales senoidales con frecuencias específicas. Este trabajo se realiza para aplicar los conocimientos adquiridos en la asignatura de Microprocesadores I, donde se abordan temas como la programación de microcontroladores, el uso de timers y la gestión de interrupciones. La creación de un generador de señales senoidales es fundamental en diversas aplicaciones electrónicas, permitiendo al estudiante experimentar con la teoría en un entorno práctico. Además, este proyecto estimula el aprendizaje de la integración de hardware y software, y fomenta habilidades en el manejo de sistemas embebidos, así como el diseño de interfaces de usuario.

Estará diseñado en varias etapas: se comenzará con la configuración del DAC para generar la señal senoidal. Posteriormente, se desarrollará la interacción mediante un teclado y se implementará la visualización de la frecuencia en un display. Para esto, se utilizarán la tarjeta de enseñanza MEPIC y la tarjeta de interfaz de teclado y pantalla, empleadas en prácticas anteriores. Finalmente, se incorporará una comunicación serie a través del USART, que permitirá la recepción de órdenes y el envío de información sobre la frecuencia actual. El método utilizado en este trabajo se basa en la programación modular (dividir en subprogramas) y el uso de interrupciones. Se implementará el Timer 0 del microcontrolador para generar interrupciones cada 62.5  $\mu$ s, lo cual se utilizará como frecuencia de muestreo para la señal senoidal. La frecuencia de salida será controlada por una variable global llamada "frec val", que podrá ser modificada mediante el teclado o a través de comandos recibidos por el puerto USART. Este enfoque garantiza que la señal se genere de forma continua e independiente de otras operaciones del sistema, como la gestión del teclado y la visualización en el display. Además, el sistema se ajustará automáticamente a cualquier cambio en la variable de frecuencia en cualquier momento del funcionamiento.

## 2 Explicación del Proyecto

El proyecto consiste en el diseño y desarrollo de un generador de señal senoidal utilizando un PIC18F45K50, con el objetivo de generar frecuencias de 125 Hz, 250 Hz y 500 Hz, que son mostradas en un display de 7 segmentos. Se implementa un sistema de muestreo y control mediante un Timer 0 que permite la producción de la señal a través de un DAC, y la interacción del usuario se realiza mediante un teclado, que valida y actualiza la frecuencia seleccionada, proporcionando confirmaciones auditivas. Además, el dispositivo envía y recibe datos a través de comunicación serial USART, respondiendo a consultas sobre la frecuencia y actualizaciones de comandos, asegurando una experiencia interactiva y eficiente. Se espera entregar un informe que detalle el desarrollo del proyecto, incluyendo diagramas de flujo y cálculos necesarios, junto con el código fuente en un archivo .zip.

## 3 Cálculos para generar las bases de tiempo

Los calculos para la configuración del tiempo(Timer 0) fueron los siguientes, con el modo de 16 bits( $T08BIT = 0$ ):

$$T = \frac{4}{F_{osc}} * (PRESCALER) * (65355 - TMR0)$$

Se uso un tiempo de 4ms, lo que el Timer0 será:

$$TMR0 = 53355$$

Se usa entonces:  $tm0h = 0xd1$  y  $tm0l = 0x1f$ . Para el espaciado entre muestras también se puede usar un `delay us()`, con los valores de 62.5ms(500Hz), 125ms(250Hz) y 250ms(125Hz).

## 4 Valores usados para la configuración del DAC

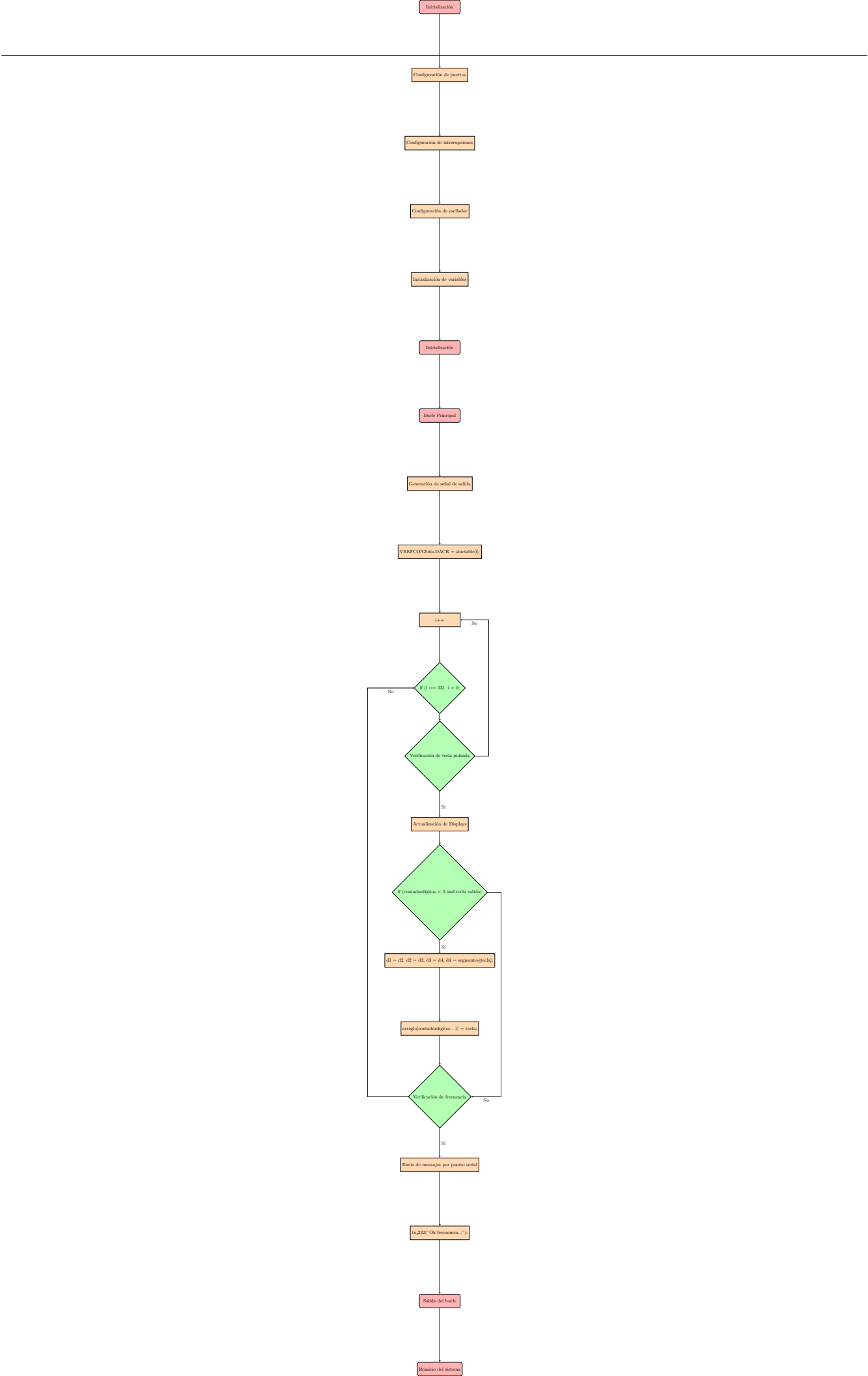
Los valores usados para el DAC fueron los siguientes:

```
//Configuración del DAC

VREFCON1bits.DACEN = 1;      //HABILITO EL DAC
VREFCON1bits.DACLPS = 1;     //FUENTE DE TENSION DE BAJA POTENCIA A Referencia positiva
VREFCON1bits.DACOE = 1;     //HABILITO TENSION DE SALIDA DEL DAC
VREFCON1bits.DACPSS = 0b10; //FUENTE POSITIVA DEL DAC
VREFCON1bits.DACNSS = 0;     //FUENTE NEGATIVA DEL DAC a VSS
VREFCON2bits.DACR = 0b1111; //Bits de selección de salida de voltaje DAC
VREFCON0bits.FVREN = 1;     //Habilito el valor de referencia ajustado
VREFCON0bits.FVRS = 0b11;   //FVRS con 4.096 Volts
VREFCON0bits.FVRST = 1;     //Salida de voltaje ajustado listo para usar
```

## 5 Diagramas de flujo

### 5.1 Main



## 5.2 Interrupcion

Inicio de Interrupción

Interrupción por recepción

Si

Leer byte recibido

¿Error de Overrun?

Reiniciar bandera de error

Incrementar contador de recepción

No

¿Coincide con F=0125?

No

Establecer contador 125 a 0

Si

Incrementar contador 125

¿Coincide con F=0250?

No

Establecer contador 250 a 0

Si

Incrementar contador 250

¿Coincide con F=0500?

No

Establecer contador 500 a 0

Si

Incrementar contador 500

¿Coincide con F=?

No

Establecer contador p a 0

Si

Incrementar contador p

¿Hay coincidencias incorrectas?

Si

Reiniciar contadores y enviar mensaje de error

¿Contador p == 3?

No

Si

Enviar frecuencia actual y mensaje

¿Contador 125 == 6?

No

Actualizar frecuencia a 125

¿Contador 250 == 6?

No

Actualizar frecuencia a 250

¿Contador 500 == 6?

No

Actualizar frecuencia a 500

Reactive bandera de interrupción

No

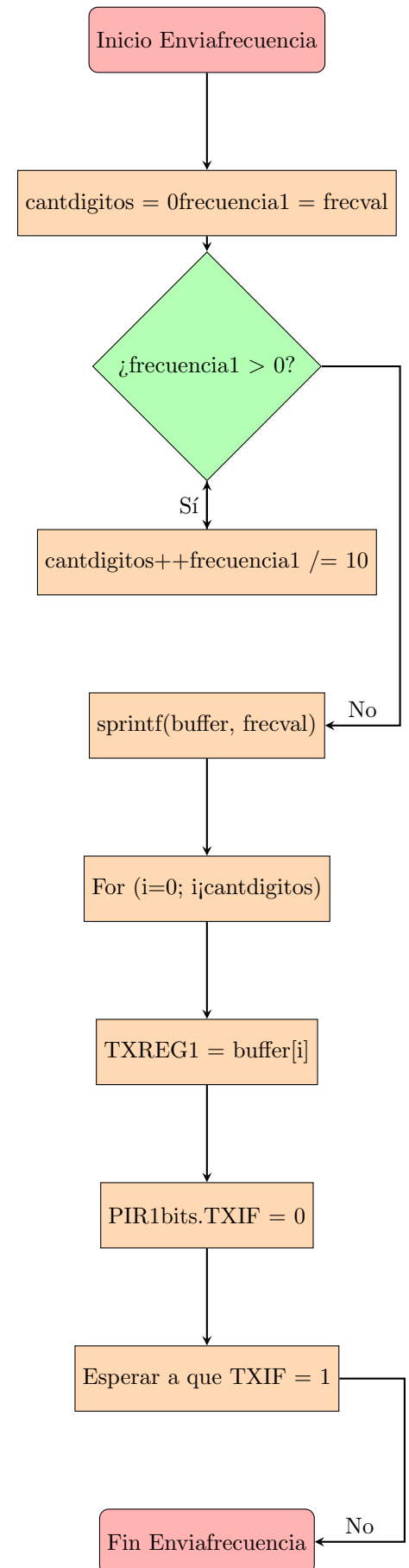
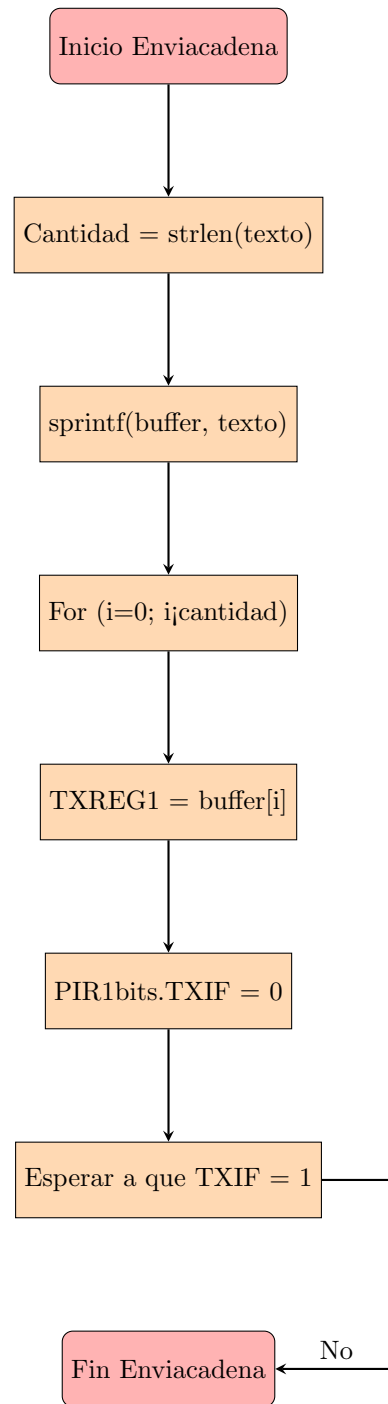
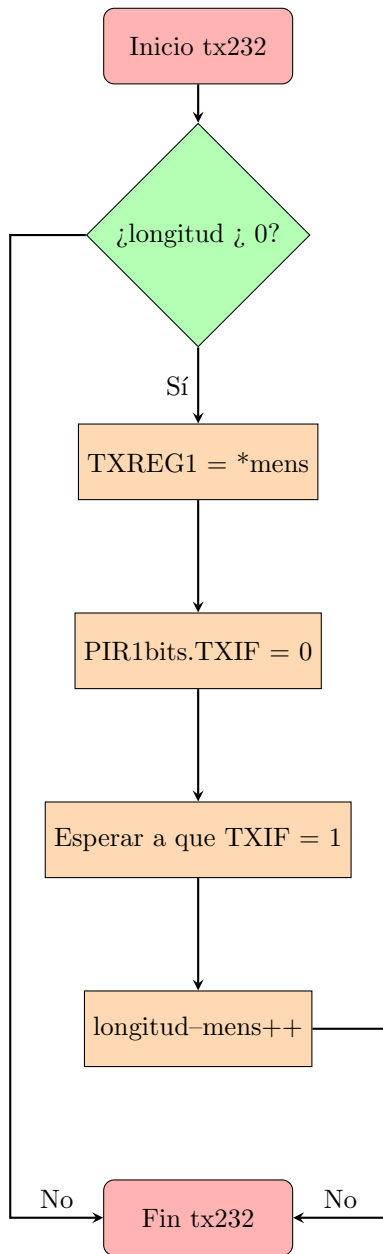
Fin de Interrupción

No





### 5.3 Comunicacion



## 6 Conclusiones

En esta práctica, se implementó un convertidor digital-analógico (DAC) utilizando un muestreo controlado por un retardo en lugar de un temporizador. Esto permitió observar cómo se generaban diferentes frecuencias de salida a partir de valores digitales, lo que es fundamental en aplicaciones de procesamiento de señales. Al utilizar un retardo para el muestreo, se pudo observar cómo la frecuencia de salida variaba en función del tiempo de espera entre cada muestreo. Esto es crucial para entender cómo se puede manipular la señal de salida en un DAC. La implementación de la comunicación serial permitió cambiar dinámicamente la frecuencia de salida del DAC. Aprendí a utilizar el teclado para enviar comandos que modifican los parámetros de la señal, lo que demuestra la importancia de la interacción entre el usuario y el sistema. La comunicación serial es esencial en sistemas embebidos, ya que permite la configuración y el monitoreo en tiempo real de los dispositivos. Las imágenes del osciloscopio muestran diferentes formas de onda generadas por el DAC. A medida que se ajustaba la frecuencia, se podía observar cómo la forma de la señal cambiaba, lo que es indicativo de la calidad del muestreo y la resolución del DAC. Las variaciones en la frecuencia de muestreo también se reflejan en la forma de la señal, lo que resalta la importancia de un muestreo adecuado para obtener una representación precisa de la señal analógica.

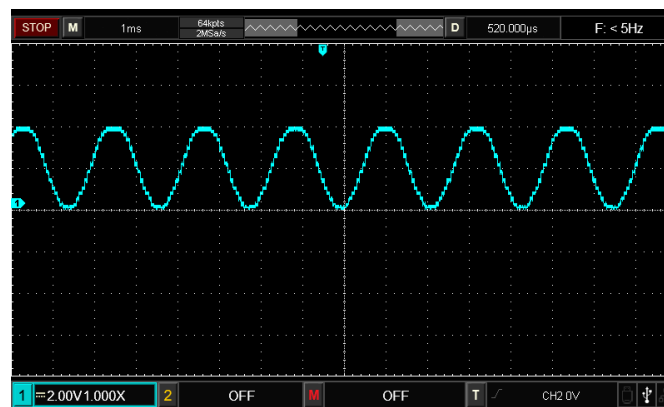


Figure 1: Señal de 500Hz

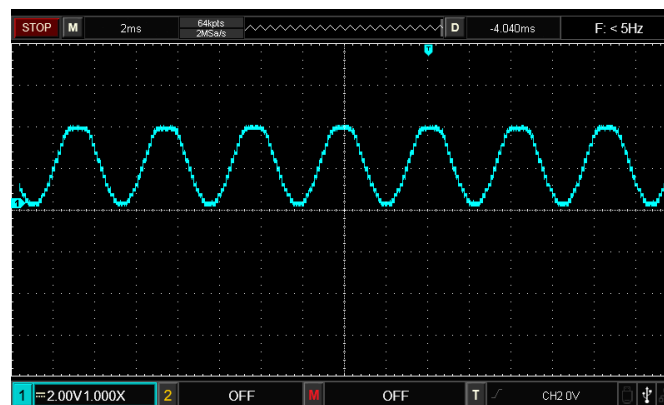


Figure 2: Señal de 250Hz

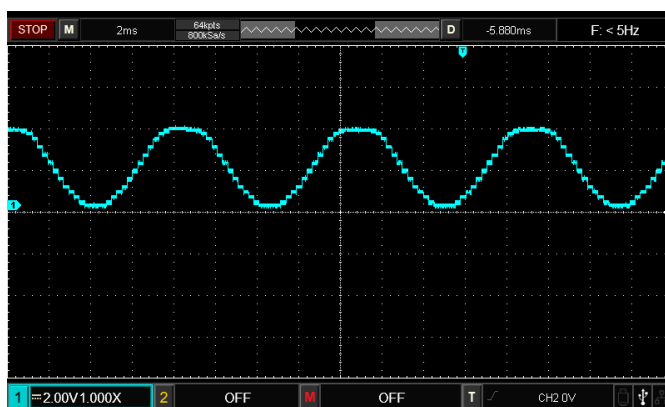


Figure 3: Señal de 125Hz