

# **SISTEMAS ELECTRÓNICOS DIGITALES**

## **TRABAJO DE MICROPROCESADORES**

Javier González Corroto 55563

Costin Leonard Matulescu 54080

Javier Tauroni Bernaldo de Quirós 53439

**Titulación:** Grado en Ingeniería Electrónica Industrial y  
Automática

**Tutor:** Luis Dávila Gómez

## ÍNDICE

<b>1 Introducción .....</b>	<b>3</b>
<b>2 Funcionalidades.....</b>	<b>3</b>
2.1 Estados del reproductor .....	3
2.2 Control de volumen .....	4
2.3 Selección de archivo .....	4
<b>3 Funcionamiento .....</b>	<b>4</b>
<b>4 Posibles mejoras.....</b>	<b>5</b>
<b>5 Conclusiones.....</b>	<b>5</b>
<b>6 Enlaces de interés.....</b>	<b>5</b>

## 1. Introducción

El objetivo de este trabajo ha sido el desarrollo de un reproductor de audio por USB utilizando la placa STM32F407VG. Entre todos los formatos de audio existentes se ha elegido que reproduzca archivos .wav, debido a su mayor calidad en comparación con otro formato más típico como es el .mp3

## 2. Funcionalidades

### 2.1. Estados del reproductor

El reproductor tiene tres estados fundamentales stop, play y pause siempre y cuando haya un USB conectado.

Inicialmente se encuentra en estado de stop donde es posible seleccionar que archivo se quiere escuchar. Al pulsar el botón de usuario pasa al estado play.

En el estado play se reproduce por la salida de audio el archivo seleccionado, al volver a pulsar el botón pasa a estado pause si es una pulsación simple o a stop si la pulsación es continua, como ocurre para apagar muchos reproductores o móviles.

En el estado pause se detiene la reproducción del audio, pero al volver a pulsar sobre el botón de usuario retoma la reproducción. Si se permanece en el estado pause durante un tiempo, unos 10s en el caso de nuestro programa, aunque también es variable, pasa a estado de stop funcionando como apagado por ahorro de energía.

Cada estado tiene asociado un LED que se enciende únicamente al estar en ese estado. El LED verde se enciende en play, el naranja en pause y el rojo en stop.

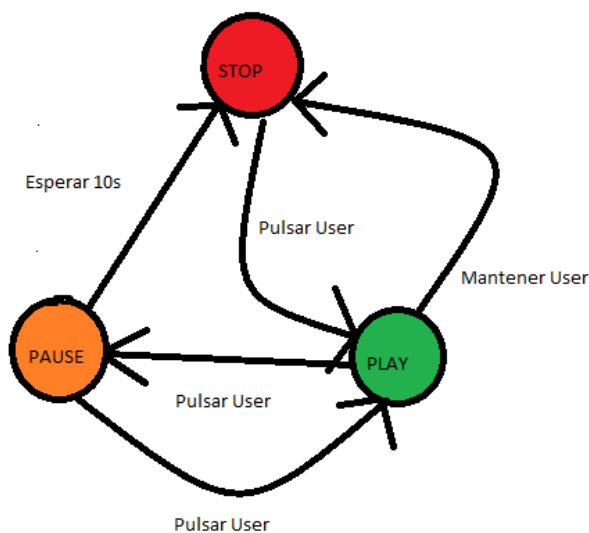


Figura 1. Diagrama de estados

### 2.2. Control de volumen

Se ha conectado un potenciómetro a la placa que permite variar el volumen de reproducción en un rango de 0 a 255. El control de volumen funciona de manera independiente, pudiéndose realizar en cualquiera de los tres estados. Si se usan cascos no es recomendable llegar a valores demasiado altos.

### 2.3. Selector de archivo

Es posible seleccionar que archivo se desea reproducir. Se puede elegir entre cinco archivos distintos conectando un cable a los pines PD8, PD9, PD10, PD11 o dejando el cable sin conectar. Es posible cambiar el cable en cualquier momento sin que ello afecte a la reproducción, sin embargo, para hacerse efectivo el cambio el programa tendrá que pasar por el estado stop. Estos archivos deberán llamarse 0.wav, 1.wav, 2.wav, 3.wav y D.wav.

### 3. Funcionamiento

Además del main.c el programa incluye otros archivos para poder realizar correctamente el programa, desde la comunicación del USB con la placa hasta la reproducción del audio. A la hora de establecer la comunicación entre el USB y la placa se ha optado por el protocolo I<sup>2</sup>C. Se ha elegido este protocolo por la posibilidad de conectar otros dispositivos utilizando el mismo bus serie en caso de ampliar las funcionalidades. Para la transmisión del audio también se ha empleado DMA.

Para mejorar la transmisión del audio se ha modificado la frecuencia de reloj a 8 MHz, por esta razón los valores usados para configurar el temporizador básico TIM2 empleado para el paso del estado pause a stop, tal y como se ha descrito en el apartado 2.1, han sido modificados en consecuencia.

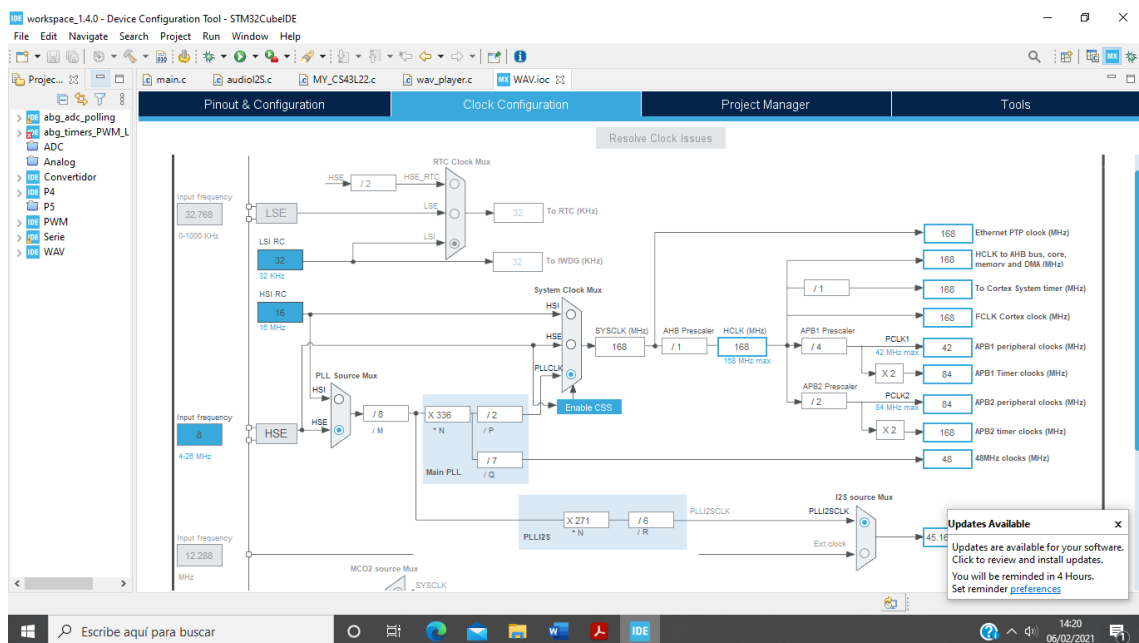


Figura 2. Cambios realizados en la frecuencia del reloj

Para evitar que el temporizador generase una interrupción instantáneamente se optó por implementar un algoritmo para ignorar esa primera interrupción instantánea.

Para que el micro lea el USB se utiliza un adaptador USB a microUSB, también se podría introducir una tarjeta SD o microSD siempre y cuando se disponga del adaptador correspondiente y sea compatible con el tipo de archivo FATFS.

Para la gestión del CS43L22, el convertidor DAC de audio que incorpora la placa se han utilizado una librería creada por M.Yaqoob.

También existe la posibilidad de conectar la salida de audio del micro a la FPGA, para usar el audio como parte del proyecto de FPGA

#### **4. Posibles mejoras**

Si bien estamos muy satisfechos con el trabajo realizado es cierto que estos proyectos siempre tienen margen de mejora, pudiéndose añadir funciones o mejorar las ya implementadas. A continuación se presentan algunas posibles mejoras que por falta de tiempo no se han desarrollado.

##### **4.1. Pantalla LCD**

Como ya se ha comentado anteriormente, la elección del protocolo I<sup>2</sup>C para las comunicaciones en serie vino motivado por la posibilidad de añadir varios periféricos compartiendo el mismo bus de datos. Uno de estos periféricos podría ser una pantalla LCD en la que se puedan visualizar datos como el nombre del archivo o el volumen.

##### **4.2. Control remoto por IR**

Otro posible añadido hubiera sido un mando a distancia que se comunicase a través de infrarrojos, gestionando así el paso de un estado a otro del reproductor. También se podría haber controlado la selección de archivo o el volumen, aunque esto último hubiera provocado que la eliminación del potenciómetro.

#### **5. Conclusiones**

Creemos que hemos realizado un buen trabajo que además nos ha servido para ampliar los conocimientos de la asignatura. El aspecto más complicado del trabajo ha sido también el más novedoso, entender el funcionamiento del audio y adaptarlo a nuestras necesidades. Aunque nos hubiera gustado disponer de algo más de tiempo para añadirle algunas de las funciones anteriormente descritas, en resumen, estamos satisfechos con los resultados del trabajo.

#### **6. Enlaces de interés**

Repositorio de donde se han obtenido las librerías para la gestión del audio:

[MYaqoobEmbedded \(github.com\)](https://github.com/MYaqoobEmbedded)

Repositorio del proyecto: [Micro/STM32 at main · Jtauroni/Micro \(github.com\)](https://github.com/Jtauroni/Micro)