

Universidad Tecnológica Nacional

Facultad Regional Tucumán



Técnicas Digitales II

Informe: Proyecto Final

Alumnos: Coronel Moreira, Carlos Javier
Palomo, José María
Romano Sanchez, Joel
Yapura, Yain Yasir

Profesor: Mansilla, Rubén Darío

Carrera: Ingeniería Electrónica

Grupo N°4

Año: 2024

Informe de proyecto Final

GENERADOR DE SEÑALES CONTROLADO VÍA BLUETOOTH

1. Consideraciones sobre el hardware del proyecto.

Descripción del proyecto

Este proyecto trata acerca del diseño de un generador de señales, utilizando la placa de desarrollo STM32, el cuál podemos manejar remotamente a través de un dispositivo que cuente con conexión bluetooth

El generador de señal está encargado de producir una señal eléctrica dependiente del tiempo, con diferentes características de amplitud, frecuencia y forma, que pueden ser fijas o variar. En nuestro proyecto podremos cambiar de señales, variando sus parámetros a través de una aplicación de un dispositivo con conexión bluetooth.

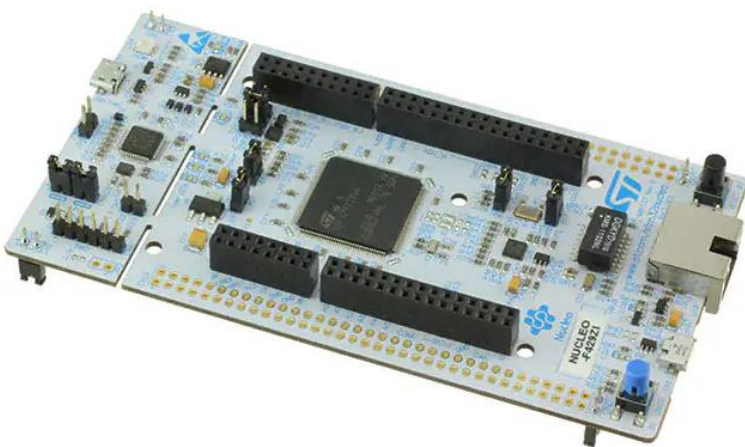
Estos parámetros se mostrarán a través de un display LCD, adaptado con módulo I2C, conectados a la placa de desarrollo.

Listado de componentes. Incluye componentes, módulos, placa de desarrollo, etc. Todo lo que constituye el hardware del dispositivo desarrollado. Debe incluir imagen de cada ítem y cantidad. La info se presenta en una tabla.

Para este proyecto usaremos los siguientes componentes:

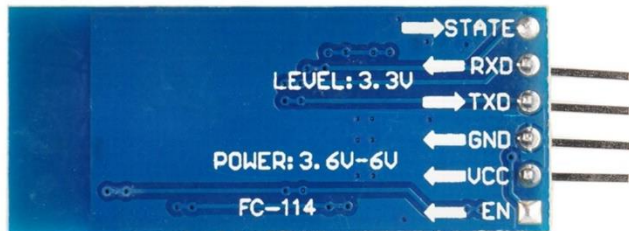
- **Placa de desarrollo: STM32-Núcleo-F429ZI**

Unidades: 1



- Módulo Bluetooth HC-06

Unidades: 1

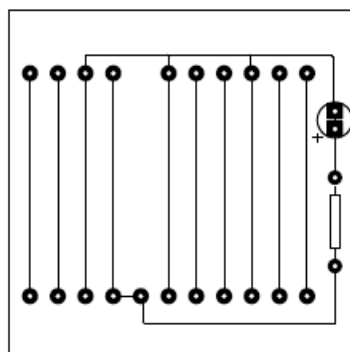


- Display LCD 16x2 con módulo I2C

Unidades: 1



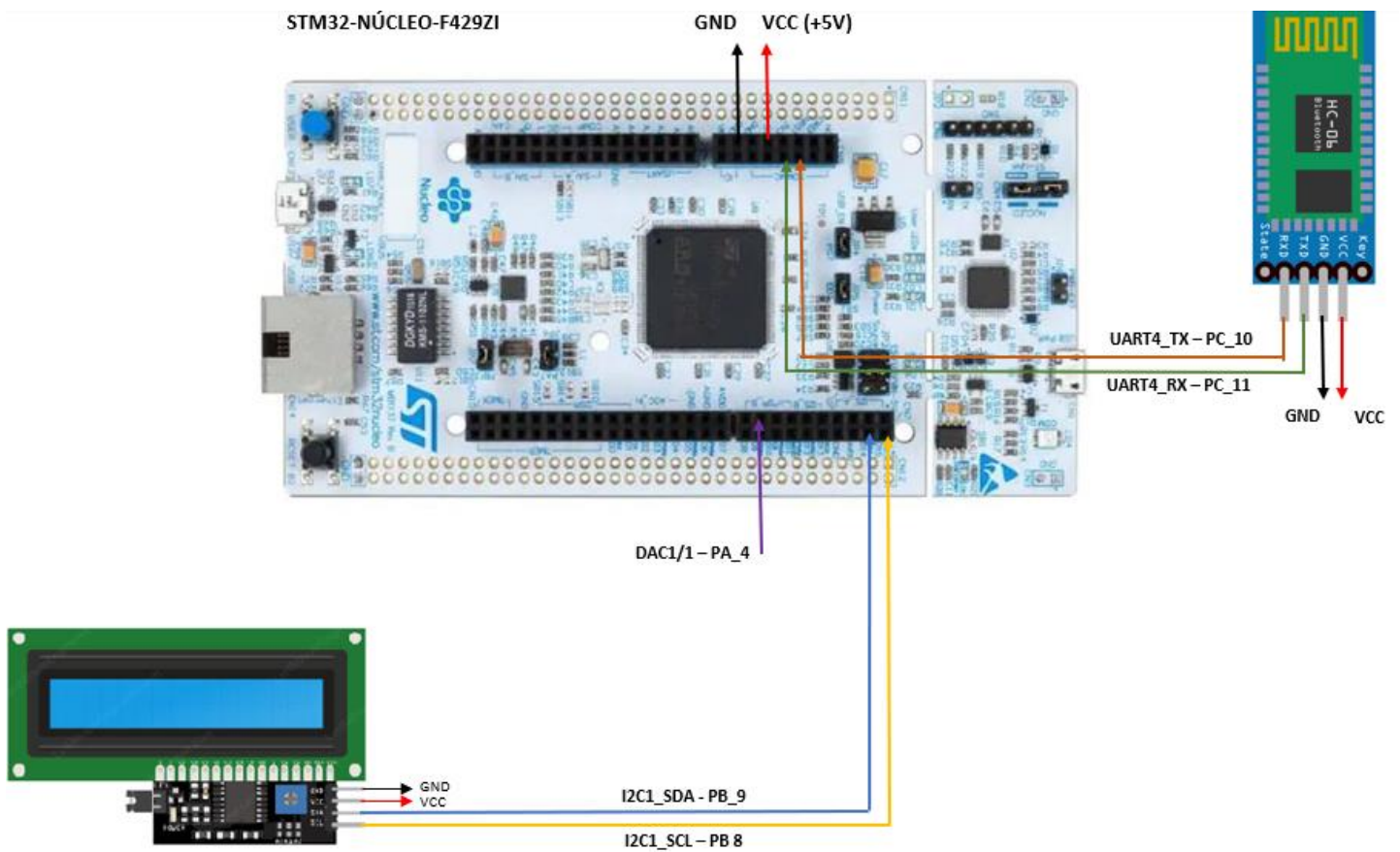
- Placa 5x5 mm



- Led RGB



- Conexión del circuito



2. Consideraciones sobre el software

Link al repositorio con el código fuente del proyecto.

<https://github.com/joelsanchez00/Grupo-4-TDII-2024/>

Nombre del Proyecto: **PROYECTO_FINAL_TDI2_GRUPO_4**

Descripción de funcionamiento de la aplicación del proyecto.

La aplicación se basa en una placa de desarrollo STM32-Nucleo-F429ZI y utiliza un módulo bluetooth HC-06 para controlar el generador de señales. Nos permite cambiar los tipos de señales, también con la posibilidad de variar sus parámetros, mostrando el tipo de señal elegida en el momento en un display LCD conectado a la placa STM mediante un módulo adaptador I2C-LCD que utiliza un driver PCF8574.

La variable **cuenta** determina el estado de la MEF. Esta variable se utiliza para seleccionar qué tipo de onda se generará (senoidal, sierra, cuadrada, triangular). Por lo tanto es una MEF de CUATRO ESTADOS.

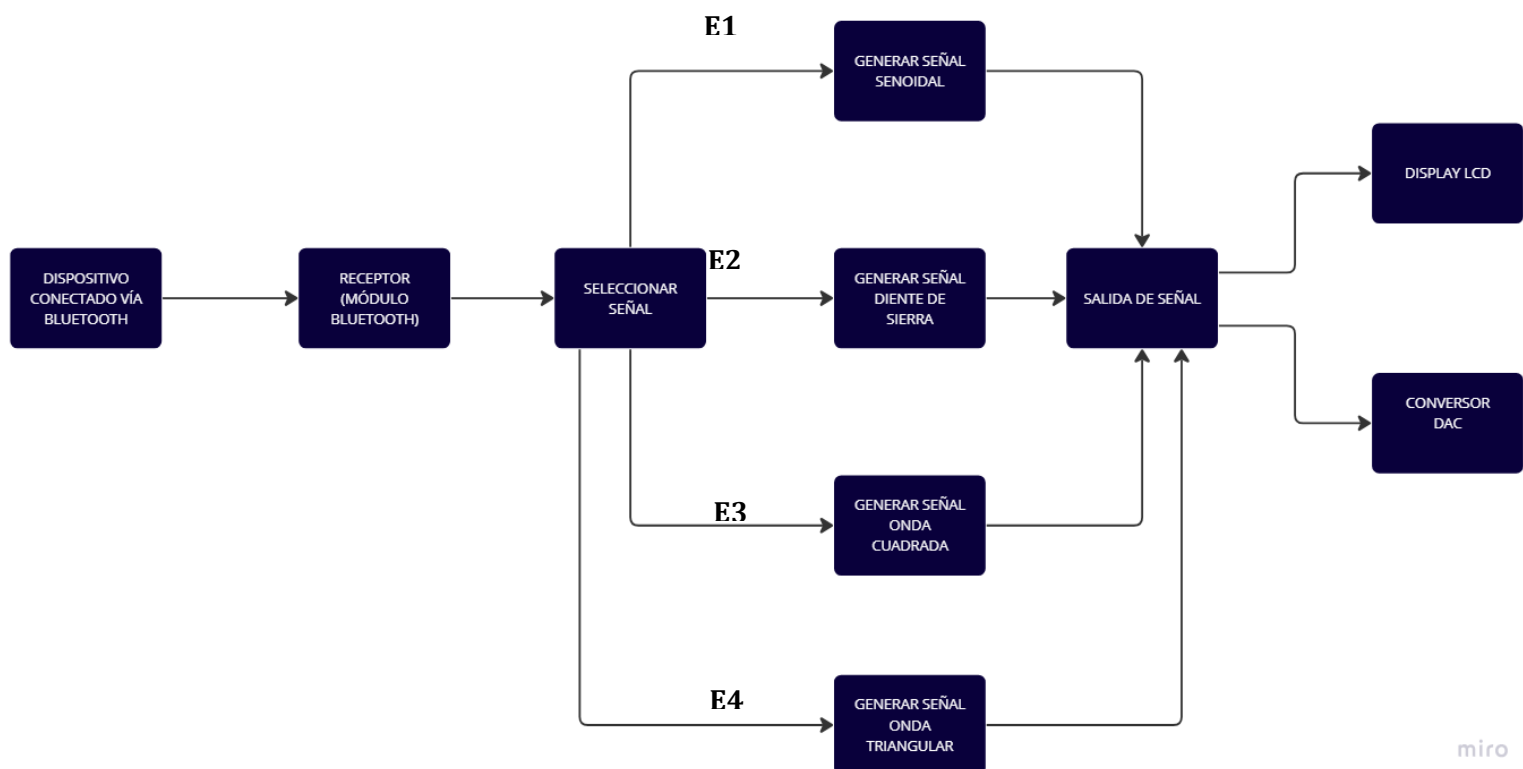
La variable **cuenta** se incrementa cuando se presiona el botón **USER_Btn_Pin**, cuando cuenta supera 4, se reinicia a 1. Las transiciones de estado son lineales, moviéndose secuencialmente entre los tipos de onda.

Los comandos bluetooth cambian la frecuencia y amplitud, pero no cambian el estado de la onda que se está generando.

Cada estado corresponde a un tipo de onda diferente.

Estados:

- **Estado 1 (SEN):** Generación de **onda senoidal**.
- **Estado 2 (SIE):** Generación de **onda sierra**.
- **Estado 3 (CUA):** Generación de **onda cuadrada**.
- **Estado 4 (TRI):** Generación de **onda triangular**.



Listado de los módulos de software desarrollados en el proyecto.

- **Librería: LCD_I2C (Driver propio)**

Utilizo en la carpeta API los siguientes módulos:

- **API_debounce:** función antirrebote. Se declaran los delays.

Listado de los periféricos que utiliza en el proyecto.

Son los siguientes:

GPIO: Para manejo de los LEDs de usuario.

I2C1: Para manejo de la comunicación con el LCD.

UART: Para manejo de la comunicación con el Módulo bluetooth HC-06

DAC: Propio de la placa, este lo usaremos como puerto de salida para nuestra señal.

3. Buenas prácticas de programación

Para crear link a una línea de código específica en un archivo que se encuentra en un repositorio siga las instrucciones del siguiente archivo: [Permalink.pdf](#)

3.1. Ejemplo de variable global privada con un link al código fuente.

https://github.com/Joelsanchez00/Grupo-4-TDII-2024/blob/a3add8f64b5eccb703e87d26da214f4089b9e6f1/PROYECTO_FINAL_TDI2_GRUPO_4/Drivers/API/Src/API_debounce.c#L44C5-L44C45

3.2. Ponga el link a la primera línea de la función de actualización de la MEF de elaboración propia.

Ubiqué el desarrollo de la MEF en **main.c** directamente por lo que paso link al inicio de la misma, en el lazo infinito:

https://github.com/Joelsanchez00/Grupo-4-TDII-2024/blob/a3add8f64b5eccb703e87d26da214f4089b9e6f1/PROYECTO_FINAL_TDI2_GRUPO_4/Core/Src/main.c#L304

3.3. Si utilizó alguna de las reglas de [Power of ten rules](#), mencionar cuál y poner el link al código fuente donde se aplica.

El código compila sin warnings, No se utilizan recursiones ni control de flujo del tipo **goto**, los lazos usados tienen límites fijos, no hay lazos que lleven a una recursividad infinita, salvo el lazo infinito de main.c

Son funciones donde se utiliza bucles For con para generar valores de onda

https://github.com/Joelsanchez00/Grupo-4-TDII-2024/blob/a3add8f64b5ecb703e87d26da214f4089b9e6f1/PROYECTO_FINAL_TDI2_GRUPO_4/Core/Src/main.c#L108

3.4. Si utilizó alguna otra regla/técnica de buenas prácticas de programación, explicar cuál y poner el link al código fuente donde se aplica.

3.5. Link a un comentario en el código que explique el por qué y **NO** el cómo:

https://github.com/Joelsanchez00/Grupo-4-TDII-2024/blob/a3add8f64b5ecb703e87d26da214f4089b9e6f1/PROYECTO_FINAL_TDI2_GRUPO_4/Core/Src/main.c#L71

4. **Anexos.**

Datasheets de componentes utilizados. Links a los datasheets: de la placa de desarrollo utilizada, de los módulos utilizados, de los sensores, etc.

Datasheet: Placa de desarrollo: STM32-Núcleo-F429ZI

<https://www.st.com/en/evaluation-tools/nucleo-f429zi.html>

Datasheet: Display LCD 16x2 con módulo I2C

https://www.waveshare.com/w/upload/4/4d/LCD1602_I2C_Module.pdf

Datasheet: Módulo Bluetooth HC-06

<https://www.alldatasheet.es/html-pdf/1179032/ETC1/HC-06/109/1/HC-06.html>

5. **Bibliografía utilizada.**

- Manual de introducción a STM32

https://wiki.st.com/stm32mcu/wiki/STM32StepByStep:Getting_started_with_STM32:_STM32_step_by_step

- "ARM® Cortex®-M4 Generic User Guide" (para núcleos Cortex-M4, como el STM32F429ZI).

- "RM0090: Reference Manual – STM32F427/437, STM32F429/439 advanced Arm®-based 32-bit MCUs"
- "PM0214: Programming manual for STM32F3 series and STM32F4 series"
- STMicroelectronics. (Año de publicación). RM0090: Reference Manual – STM32F427/437, STM32F429/439 advanced Arm®-based 32-bit MCUs.