

# Generación y validación de un Token con stm32

Integrantes: Cesar,  
Manuel,  
Neyder Alberto Vargas Guerrero

**Resumen**—En este informe se detalla el diseño y la implementación de un sistema de generación de tokens mediante funciones hash utilizando un microcontrolador STM32F411 y una pantalla LCD 16x2. La generación del token se realiza mediante la configuración de timers en el microcontrolador para garantizar la precisión del tiempo, y se emplean funciones hash para combinar datos clave y producir un token único y seguro. La interfaz con la pantalla LCD permite visualizar el token generado y proporcionar retroalimentación al usuario. Además, se implementa un programa en C para sincronizar con el microcontrolador, recibir el token generado y validar su autenticidad utilizando un algoritmo específico. Las pruebas realizadas confirman la precisión en la generación de tokens y la eficacia en su validación, destacando la robustez y confiabilidad del sistema en entornos que requieren altos niveles de seguridad.

**Keywords**— Control token, función hash, Microcontrolador STM32, Display LCD 16x2, programación en C

## I. INTRODUCCIÓN

En el ámbito de los sistemas embebidos, la seguridad de la información es un aspecto crucial que requiere constantes innovaciones y desarrollos. En este contexto, este informe describe un laboratorio realizado como parte del curso de sistemas embebidos, que se centra en la implementación de un sistema de generación de tokens y su validación utilizando el microcontrolador STM32F411 y una pantalla LCD 16x2.

El objetivo principal de esta investigación es explorar y aplicar conceptos teóricos en un entorno práctico, con énfasis en la seguridad y la eficiencia de los sistemas embebidos.

El informe detallará los pasos seguidos en la configuración del microcontrolador STM32F411, incluida la programación de timers para la generación precisa de tokens y la interfaz con la pantalla LCD 16x2 para la retroalimentación visual. Además, se abordará la aplicación de funciones hash para garantizar la unicidad y seguridad de los tokens generados, así como la implementación de un programa en C para la sincronización y validación del token.

A través de este proyecto se espera que los resultados obtenidos contribuyan al avance del conocimiento en esta área y sirvan como base para futuras investigaciones y aplicaciones prácticas en el campo de la tecnología embebida.

## II. MARCO TEORICO

El microcontrolador STM32F411, parte de la familia STM32 de STMicroelectronics, es una solución versátil para aplicaciones embebidas. Basado en la arquitectura Cortex-M4 de ARM, ofrece un equilibrio entre potencia de procesamiento y eficiencia energética. Equipado con periféricos integrados como interfaces de comunicación, timers y convertidores ADC, facilita la implementación de diversas funciones. Con memoria flash para almacenamiento de programas y RAM para datos temporales, proporciona capacidad de almacenamiento adecuada. Su soporte para el entorno de desarrollo STM32CubeIDE simplifica la programación. En resumen, el STM32F411 es una opción popular debido a su flexibilidad, rendimiento y facilidad de desarrollo en aplicaciones embebidas.

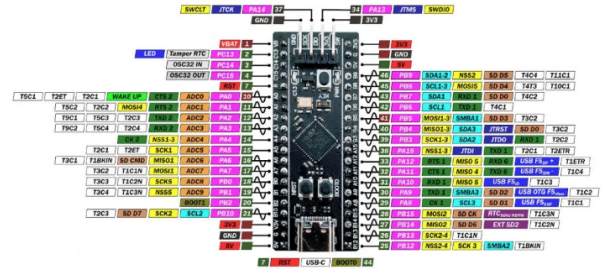


Figura 1. Placa STM32F411CEU6

La fotoresistencia, también conocida como LDR (Light Dependent Resistor) o fotorresistor, es un componente electrónico cuya resistencia varía en función de la intensidad de la luz incidente sobre ella. Cuando la luz incide sobre la fotoresistencia, la resistencia disminuye, y viceversa en ausencia de luz. Este comportamiento la hace útil en una variedad de aplicaciones, como circuitos de control de iluminación automática, seguidores de luz solar, sistemas de detección de luminosidad, entre otros. La fotoresistencia es un componente pasivo y su sensibilidad a la luz puede variar según el material semiconductor utilizado en su fabricación. Su uso se ha popularizado en electrónica debido a su simplicidad y versatilidad en diversas aplicaciones.

### Especificaciones:

- **Rango de Resistencia:** La resistencia varía desde varios ohmios en la oscuridad hasta varios megaohmios en la luz brillante.
- **Sensibilidad Espectral:** La sensibilidad a diferentes longitudes de onda de la luz puede variar.
- **Respuesta a la Luz:** La respuesta a cambios en la intensidad de la luz puede variar en términos de velocidad y linealidad.
- **Tiempo de Respuesta:** El tiempo que tarda la fotoresistencia en cambiar su resistencia en respuesta a cambios en la luz.
- **Temperatura de Funcionamiento:** La temperatura a la que puede operar sin degradación significativa en su rendimiento.
- **Potencia Máxima:** La potencia máxima que puede disipar sin dañarse.
- **Tamaño y Forma:** Las dimensiones físicas de la fotoresistencia.

### Aplicaciones:

- **Control de Iluminación Automática:** Utilizada en circuitos que ajustan la iluminación en función de la cantidad de luz ambiente.
- **Seguidores de Luz Solar:** Implementada en sistemas que orientan dispositivos hacia la posición óptima para recibir luz solar.
- **Sistemas de Detección de Luminosidad:** Empleada en aplicaciones que detectan la presencia o ausencia de luz en un entorno específico.
- **Sensores de Oscuridad:** Utilizada en dispositivos que activan o desactivan funciones en función del nivel de luz presente.
- **Control de Pantallas Retroiluminadas:** Implementada en dispositivos que ajustan automáticamente la intensidad de la retroiluminación de las pantallas según la luz ambiente.

- **Control de Velocidad de Impresoras:** Empleada en impresoras láser para ajustar la velocidad del proceso de impresión en función de la luz ambiente.
- **Sistemas de Seguridad:** Utilizada en sistemas de seguridad que activan alarmas o luces de emergencia en condiciones de poca luz.

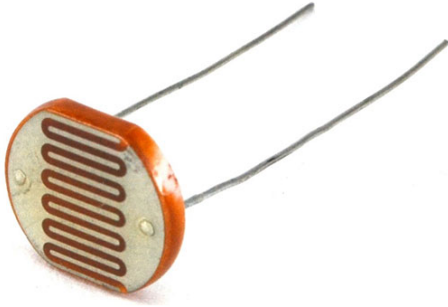


Figura 2. Sensor Fotoresistor

Una función hash es un algoritmo matemático que toma una entrada de datos de longitud variable y la convierte en una cadena de longitud fija, generalmente de tamaño mucho menor. Esta cadena resultante se conoce como "hash." resumenz es única para cada conjunto de datos de entrada. Las funciones hash tienen varias propiedades importantes:

1. **Determinismo:** Dada la misma entrada, una función hash siempre producirá la misma salida.
2. **Eficiencia:** El cálculo del hash debe ser rápido y eficiente para ser útil en aplicaciones prácticas.
3. **Unicidad:** Cada conjunto de datos de entrada debe tener un hash único asociado, pero diferentes entradas pueden producir el mismo hash (colisión) debido a la reducción de la longitud de salida.
4. **Avalancha:** Un pequeño cambio en los datos de entrada debe producir un cambio significativo en el hash de salida.

Las funciones hash se utilizan en una amplia variedad de aplicaciones, como la criptografía, la integridad de datos, la identificación de duplicados, la indexación y la búsqueda de datos, entre otros. Algunos ejemplos comunes de funciones hash son SHA-256, MD5 y SHA-1, aunque este último se considera inseguro para aplicaciones críticas debido a vulnerabilidades conocidas.

La LCD (Liquid Crystal Display) es un tipo de pantalla que utiliza cristales líquidos para producir imágenes. Funciona a partir de la modulación de la luz a través de un material cristalino que tiene la capacidad de cambiar su orientación bajo la influencia de un campo eléctrico.

1. **Capas de la LCD:** Una pantalla LCD está compuesta por varias capas, incluyendo dos sustratos de vidrio transparente, entre los cuales se encuentran depositadas las capas de material conductor y el material de cristal líquido.
2. **Material de Cristal Líquido:** El material de cristal líquido se encuentra entre las capas de sustrato de vidrio. Este material tiene la propiedad de alinearse en una dirección específica cuando se somete a un campo eléctrico.
3. **Electrodos:** En las capas de vidrio se depositan patrones de electrodos, que se utilizan para aplicar el campo eléctrico al material de cristal líquido. Estos electrodos están dispuestos en una matriz que define los píxeles individuales de la pantalla.
4. **Polarizadores:** En la parte delantera y trasera de la pantalla LCD se colocan polarizadores que permiten controlar la orientación de la luz que pasa a través de la pantalla.

5. **Modulación de la Luz:** Cuando se aplica un voltaje a los electrodos, se genera un campo eléctrico que modifica la orientación de las moléculas de cristal líquido entre los electrodos. Esto cambia la forma en que la luz polarizada se refleja o se transmite a través de la pantalla, lo que permite controlar la intensidad y el color de cada píxel.
6. **Controlador:** Un controlador de pantalla se encarga de enviar los datos de imagen a los electrodos de la pantalla, lo que permite mostrar imágenes y gráficos en la LCD.

En resumen, una LCD funciona modulando la orientación de cristales líquidos entre capas de vidrio transparente para controlar la transmisión de luz y producir imágenes. Este principio de funcionamiento permite que las pantallas LCD sean delgadas, livianas y consuman menos energía en comparación con otros tipos de pantallas, lo que las hace ampliamente utilizadas en dispositivos electrónicos como teléfonos móviles, computadoras portátiles, televisores y relojes inteligentes, entre otros.



Figura 3. Pantalla LCD

### III. DIAGRAMA DE FLUJO

El diagrama de flujo se concentra en el paso a paso de la generación de token mediante el programa de la stm.

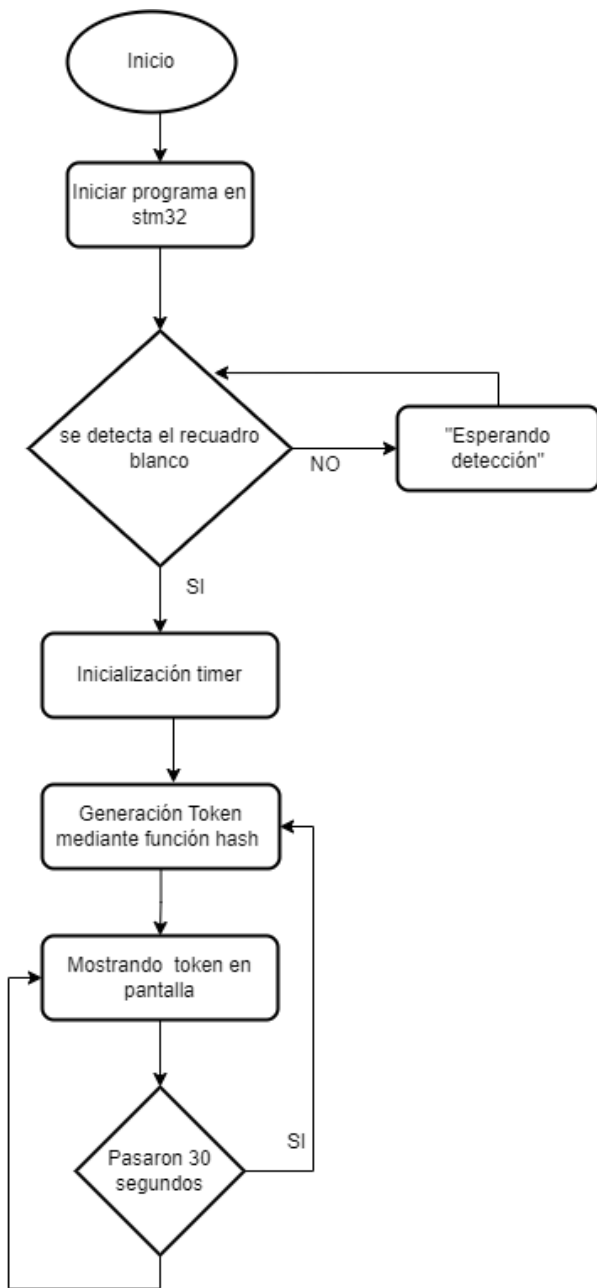


Figura 4. Diagrama de flujo – Generación token

#### IV. PROCEDIMIENTO

Para realizar el laboratorio seguimos los siguientes pasos en la codificación del primer programa generador del token:

- **Configuración Inicial:** El programa comienza configurando la entrada GPIOA0 para recibir la señal de entrada de la fotoreistencia, la cual está conectada en serie con una resistencia de 5k ohmios. Esta configuración se realiza en el código del microcontrolador, asignando GPIOA0 como una entrada digital.
- **Detección de la Señal:** El programa monitorea constantemente el estado de la señal en la entrada GPIOA0. Cuando se detecta una señal, el programa procede a verificar si su valor es igual a 0.
- **Proceso de Verificación:** Si la señal es igual a 0, el programa permanece dentro de un ciclo esperando a que la señal cambie. Esto puede indicar que se ha detectado un cambio en la

iluminación ambiental o la presencia de un objeto que interfiere con la señal de la fotoreistencia.

- **Generación del Token:** Cuando la señal detectada en GPIOA0 no es igual a 0, el programa ejecuta el proceso de generación del token. Esto implica el uso de una clave predefinida y una función hash para calcular un valor único, que actúa como el token. Este token puede ser utilizado para identificar y autenticar al dispositivo o usuario.
- **Visualización en la Pantalla LCD:** Una vez generado el token, el programa lo muestra en la pantalla LCD. Esto se logra utilizando funciones de control de la pantalla LCD para escribir el token en la pantalla. El token puede mostrarse junto con otra información relevante, como la fecha y la hora de generación.
- **Intervalo de Tiempo:** Después de mostrar el token en la pantalla LCD, el programa espera un intervalo de tiempo de 30 segundos antes de volver a verificar la señal de la fotoreistencia. Durante este tiempo, el programa puede realizar otras operaciones o simplemente esperar inactivo.

En la implementación del segundo código, se tiene un código de muestra dado por el profesor en el cual se puede ver que hay una definición de funciones y variables que ayudan a la creación del recuadro en pantalla, así mismo como la definición de funciones que permiten la obtención del tiempo para contabilizar una vez se ha hecho la sincronización entre los dos programas y así decodificar el token ingresado y ver el tiempo que está corriendo en ese instante en el otro programa.

Pero para ello era necesario crear una comparación y definir un rango en el cual el valor decodificado del token fuera válido. Pensamos en restar los valores de tiempo obtenidos, en este caso el que está corriendo en el programa en C normal y el tiempo del token ingresado, y definir que si esa resta no era mayor a los 30 segundos, entonces el token era válido y mostraría el valor del token y el tiempo.

Pero ahí es donde obtuvimos problemas al ejecutar esa lógica, ya que aparecían algunos valores raros y no coincidían con las validaciones que se querían obtener. Quizás el rango que estábamos utilizando o la forma de comparación no eran correctas.

#### V. CONCLUSIONES

- **Desafíos en la Implementación:** La configuración inicial del programa para la recepción de señales de la fotoreistencia y la generación del token presentó desafíos técnicos. Fue necesario definir funciones y variables adecuadas para garantizar un funcionamiento correcto.
- **Sincronización y Decodificación del Token:** La sincronización del tiempo entre los programas y la decodificación del token ingresado mostraron ser aspectos críticos del sistema. A pesar de la implementación de un rango de tiempo para validar los tokens, surgieron dificultades en la comparación de valores y la detección de valores válidos.
- **Posibles Mejoras:** Para mejorar la precisión y la fiabilidad del sistema, necesitamos revisar la lógica de comparación de tiempos y la definición del rango de tiempo para la validación de los tokens.
- **Lecciones Aprendidas:** La implementación de sistemas de autenticación y sincronización de este tipo presenta desafíos únicos que requieren de verdad de conocer los conceptos esenciales de los sistemas embebidos.



### VI-B. Diagrama de flujo

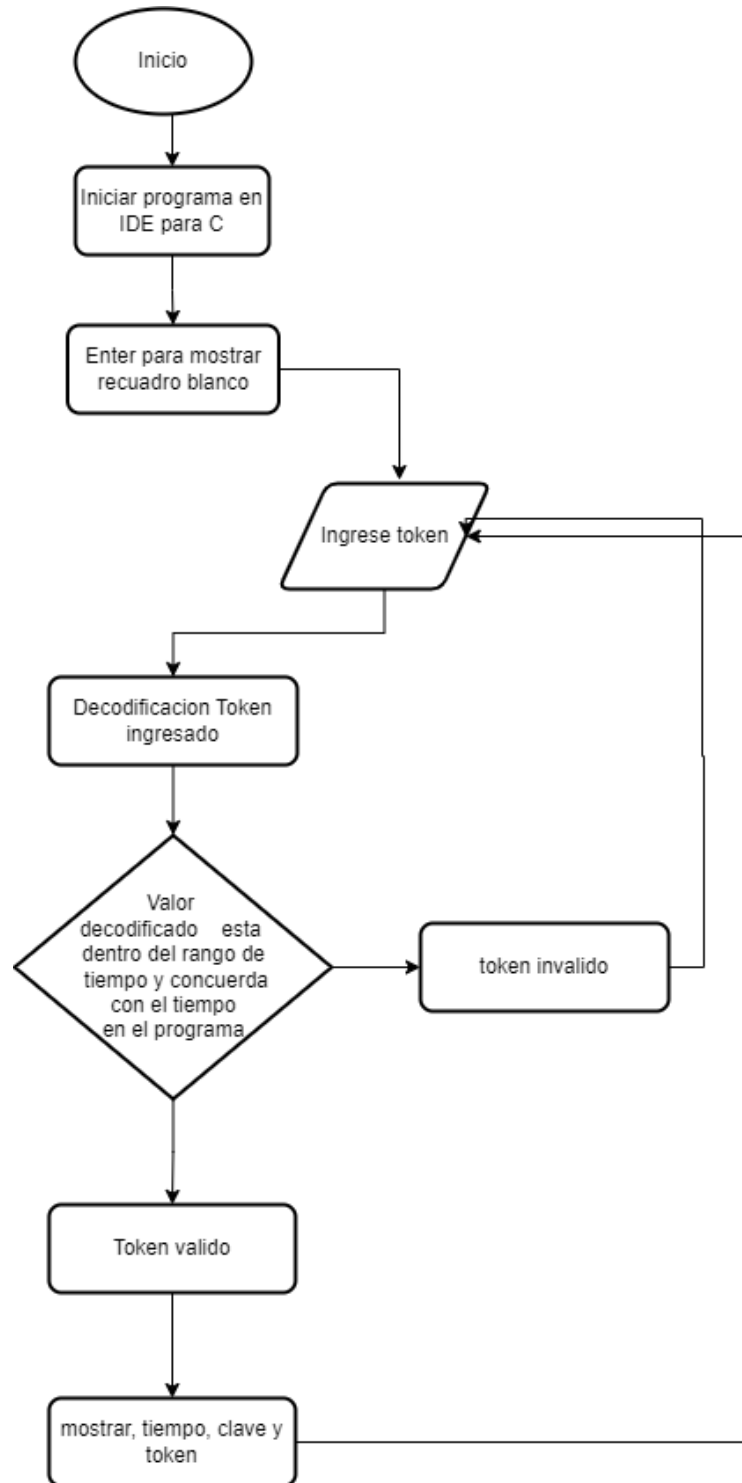


Figura 6. Diagrama de flujo decodificación token

### VII. REFERENCIAS

[1]Mischianti, R. 2022, 7 junio. WeAct STM32F411CEU6 Black-Pill: high-resolution pinout and specs. Renzo Mischianti. <https://mischianti.org/weact-stm32f411ceu6-black-pill-high-resolution-pinout-and-specs/>