Sistema de verificación de TOKEN numérico con tecnología OTP

Integrantes: Monica Julieth Paez,
Lizeth Camila Sánchez

***Resumen*—Este trabajo presenta un sistema de verificación de TOKEN numérico basado en tecnología OTP (contraseña de un solo uso). El sistema se implementa utilizando un programa en C y un sistema embebido con un sensor de luz visible y una pantalla LCD.**

***Keywords*—** Token, TIM, Fotodiodo, XOR, OTP, pantalla LCD

I. Introducción

En el contexto actual, la seguridad informática es un tema de vital importancia. La autenticación de usuarios es un componente fundamental para proteger sistemas y datos contra accesos no autorizados. Las contraseñas tradicionales son una forma común de autenticación, pero presentan vulnerabilidades como la reutilización de contraseñas o el robo de las mismas.

Los objetivos principales son, implementar un programa en C que genere un recuadro blanco en pantalla, inicie un cronómetro en milisegundos y permita consultar el valor del cronómetro en cualquier momento, desarrollar un sistema embebido que detecte el recuadro blanco mediante un sensor de luz visible, sincronice su cronómetro con el del programa en C y calcule el TOKEN cada 30 segundos realizando una operación XOR bit a bit entre el valor del cronómetro y una clave secreta, mostrar el TOKEN en la pantalla LCD del sistema embebido y enviarlo al programa en C para su verificación, por último implementar en el programa en C la lógica para verificar la validez del TOKEN recibido del sistema embebido, realizando la operación XOR inversa con la misma clave utilizada para su cálculo.

El informe del laboratorio se estructura de la siguiente manera. En primer lugar, se tiene una descripción del contexto, la tecnología OTP y la motivación del trabajo, luego se revisan en detalle de los métodos y herramientas utilizados para el desarrollo del sistema, con una explicación detallada de los códigos usados y se presentan los resultados obtenidos durante las pruebas del sistema.

II. Marco teorico

La placa STM32F401CEUx es una plataforma de desarrollo de microcontroladores de la serie STM32 de la empresa STMicroelectronics. Los microcontroladores de esta serie se basan en la arquitectura ARM Cortex-M y son conocidos por su alta eficiencia energetica y su alto rendimiento. El microcontrolador STM32F401CEU6 que se encuentra en la placa cuenta con una arquitectura de procesador ARM CortexM4 de 32 bits y una velocidad de reloj de hasta 84 MHz. Esta arquitectura permite un alto rendimiento de procesamiento de datos y la posibilidad de realizar tareas complejas en tiempo real[1]

La placa ofrece una amplia gama de características, incluyendo interfaces de comunicacion como USB, UART, SPI y I2C. Estas interfaces permiten la comunicacion con otros dispositivos y la transmision de datos. Además, la placa cuenta con una gran cantidad de pines GPIO que permiten la conexión de sensores y actuadores, lo que permite la implementación de proyectos de electronica y robótica, también cuenta con una memoria Flash de 512 KB y SRAM de 128 KB, lo que permite el almacenamiento de programas y datos. Esto es especialmente importante en proyectos donde se requiere una gran cantidad de memoria para el procesamiento de datos complejos y la implementacion de algoritmos de control.

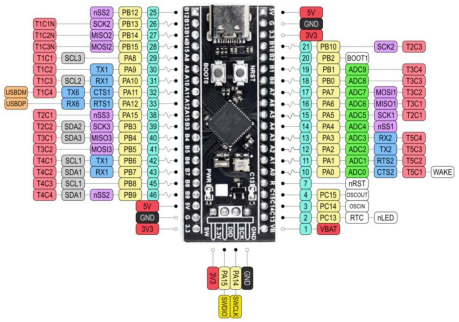

Figura 1. Placa STM32F401CEUx

XOR es una operación lógica que toma dos entradas binarias y produce una salida de 1 si el número de unos en las entradas es impar; de lo contrario, genera 0. Simbolizado por el símbolo $\oplus$, XOR puede considerarse .^{ex}clusivo"porque excluye la posibilidad de que ambas entradas sean 1. [2]

Una función criptográfica hash- usualmente conocida como "hash"- es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud.[3]

La tecnología OTP (contraseña de un solo uso) se presenta como una alternativa más segura a las contraseñas tradicionales. Se basa en la generación de claves únicas y válidas por un tiempo limitado, lo que reduce el riesgo de ataques y accesos no autorizados.

Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Puede también ser llamado fotorresistor, fotoconductor, célula fotoeléctrica o resistor dependiente de la luz, cuya siglas, LDR, se originan de su nombre en inglés light-dependent resistor. Los valores de la resistencia para estos dispositivos varían dependiendo del uso que le demos y la luz disponible, los valores típicos varían entre 1 M Ω, o más, en la oscuridad y 100 Ω con luz brillante.[4]

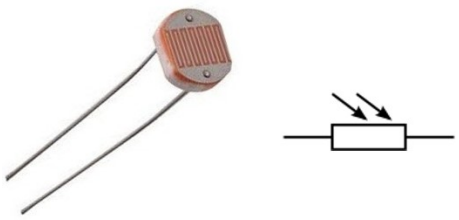

Figura 2. Fotoresistencia

Por otro lado, con el propósito de facilitar el manejo de los cricuitos se realiza un proceso de conversión Analógico-Digital

(ADC), mediante el cual se convierte una magnitud física como un voltaje, corriente, temperatura, etc. en un número binario (o señal digital).[5]

El ADC de 12-bits es un convertidor analogico-digital de aproximación sucesiva. Tiene hasta 19 canales multiplexados, permitiendo medir señales de hasta 16 fuentes externas, dos internas, y el canal VBAT. La conversión A/D de los canales puede ser hecha en modo único, continuo, de escaneo o discontinuo. El resultado es almacenado en un registro de 16-bit.

III. Diagrama de flujo

El primer diagrama de flujo se concentra en el orden general de ambos códigos, ya que en un principio se genera el recuadro blanco, el cual es detectado por el foto diodo y lleva a la sincronización e inicialización del timer, el cual cada 30 segundos cambia el valor del toke, el cual se puede consultar en cualquier instante de tiempo y se muestra en la pantalla lcd, de modo tal que el usuario pueda escribir el token en el computador, seguido se verifica el token con la tecnología HASH que utiliza la operación lógica XOR

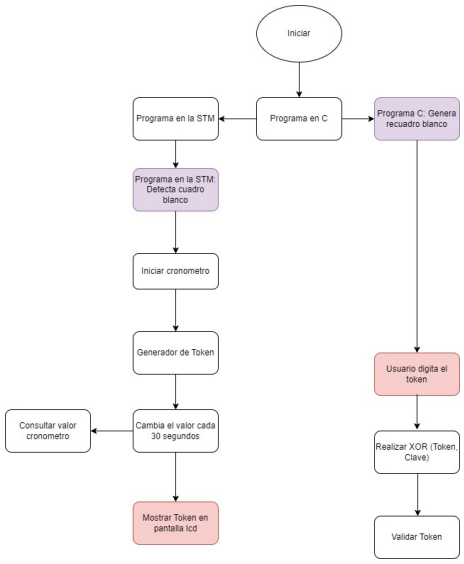

Figura 3. Diagrama de flujo general de los códigos

Este segundo diagrama de flujo se centra en la dinámica del código y las secuencias y condiciones que se usan.

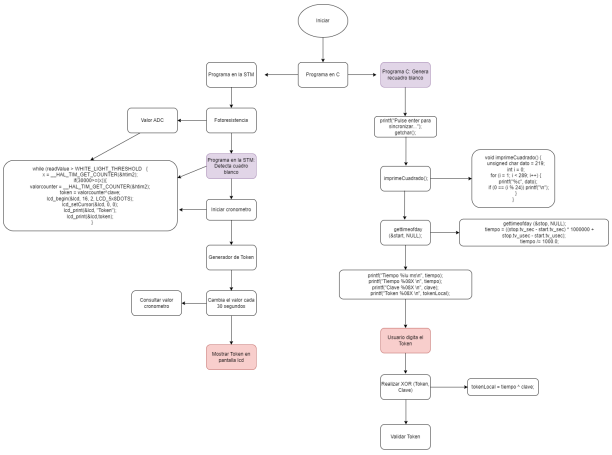

Figura 4. Diagrama de flujo hacia el código

IV. Procedimiento

El primer código en C rastrea el tiempo entre pulsaciones de teclas del usuario y genera un identificador único basado en esta medición.

Comienza pidiendo al usuario que presione Intro para sincronizar y luego imprime un cuadrado en la consola. La función printSquare() simplemente imprime un patrón de bloques sólidos en la consola para darle al programa una representación visual.

```
printf("Pulse enter para sincronizar...");
getchar();
imprimeCuadrado();
gettimeofday (&start, NULL);
```

Luego ingresa a un bucle infinito donde espera que el usuario ingrese "TOKEN". Cada vez que el usuario ingresa una etiqueta, el programa mide el tiempo desde el inicio del ciclo usando la función gettimeofday(), que proporciona una precisión de microsegundos.

```
while(1){
printf("Ingrese su TOKEN >>>");
getchar();
gettimeofday (&stop, NULL);
tiempo =((stop.tv_sec - start.tv_sec)*
1000000 + +stop.tv_usec - start.tv_usec);
tiempo/= 1000.0;
printf("\n");
```

Dentro de una función, realizamos una operación XOR bit a bit entre el tiempo medido y una clave predefinida de 4 bytes para generar un identificador único. Este ID se imprime en hexadecimal junto con el tiempo transcurrido y la clave.

```
void calculatoken()
{
token = tiempo^clave;

lcd_begin(&lcd, 16, 2, LCD_5x8DOTS);
lcd_setCursor(&lcd, 0, 0);
lcd_print(&lcd, "Token: ");
lcd_setCursor(&lcd, 0, 1);
sprintf(texto, "%X", token);
lcd_print(&lcd,texto);

}
```

En el segundo código, el que se ejectua en el ide del sistema embebido, se tiene que:

```
 HAL_TIM_Base_Start_IT(&htim2);

while(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_1)== 1);

  __HAL_TIM_SET_COUNTER(&htim2,0);

  while (1)
  {

  valorcounter =(uint32_t)
  __HAL_TIM_GET_COUNTER(&htim2);

  x = (uint32_t)valorcounter - tiempo;
   if(60000 <= x)
   {

       //__HAL_TIM_SET_COUNTER(&htim2,0);
      tiempo = __HAL_TIM_GET_COUNTER(&htim2);
      calculatoken();
```

```
    }
}
```

El timer 2 es diferente es de unit32_t Counter period es de 4 bytes

V. Conclusiones

- Durante la realización del sincronizado, se tuvieron varios problemas para la visualización del hexadecimal, y su cantidad de bytes, ya que, se estaba declarando en variables pequeñas que no permitian realizar bien la operación y por lo tanto no se visualizaba bien.
- Al obtener perfectamente el token en la STM, pasamos a la validación del token mediante el visual, en donde no se logro tener una validación satisfactoria, ya que, los tiempos en los que se tienen pareciera estar desfasados y aunque se colocabara un rango no llegaba al estar perfectamente sincronizado.

VI. Anexos

VI-A. Esquematico

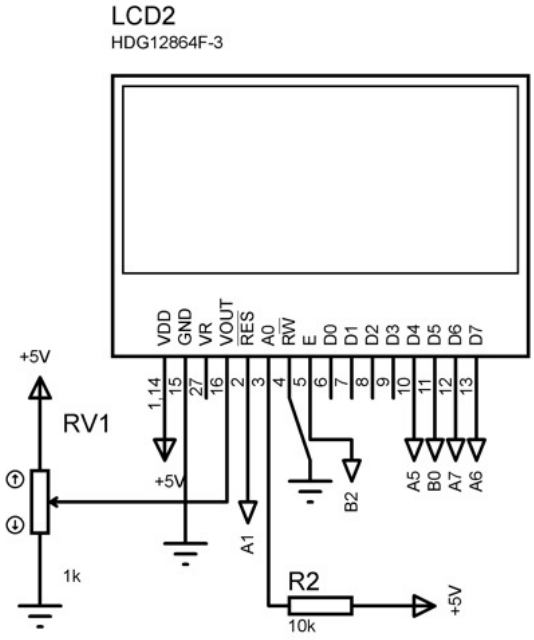

Figura 5. Esquemático conexión de la lcd

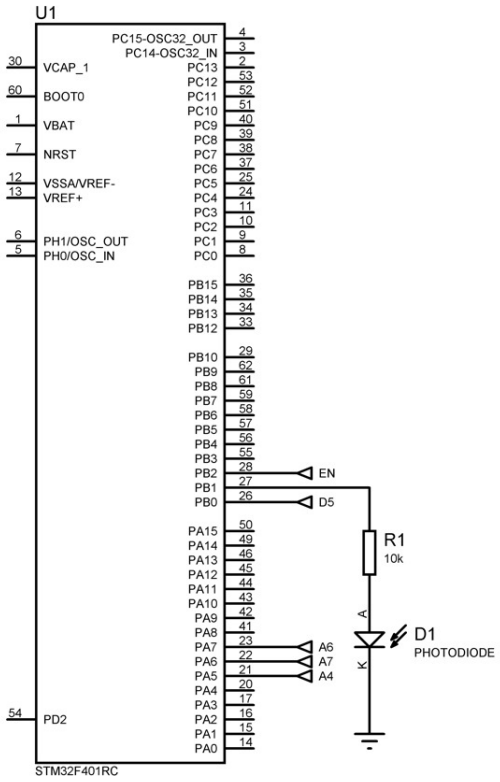

Figura 6. Esquemático stm32

VI-B. Diagrama de flujo

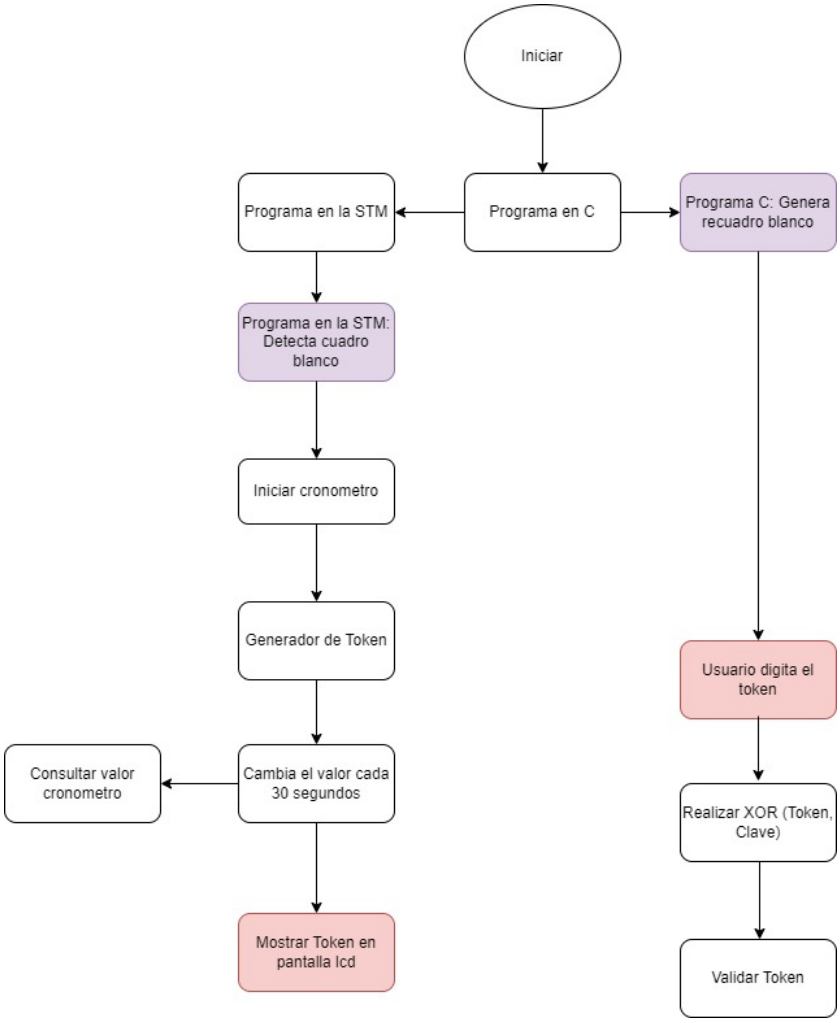

Figura 7. Diagrama de flujo general de los códigos

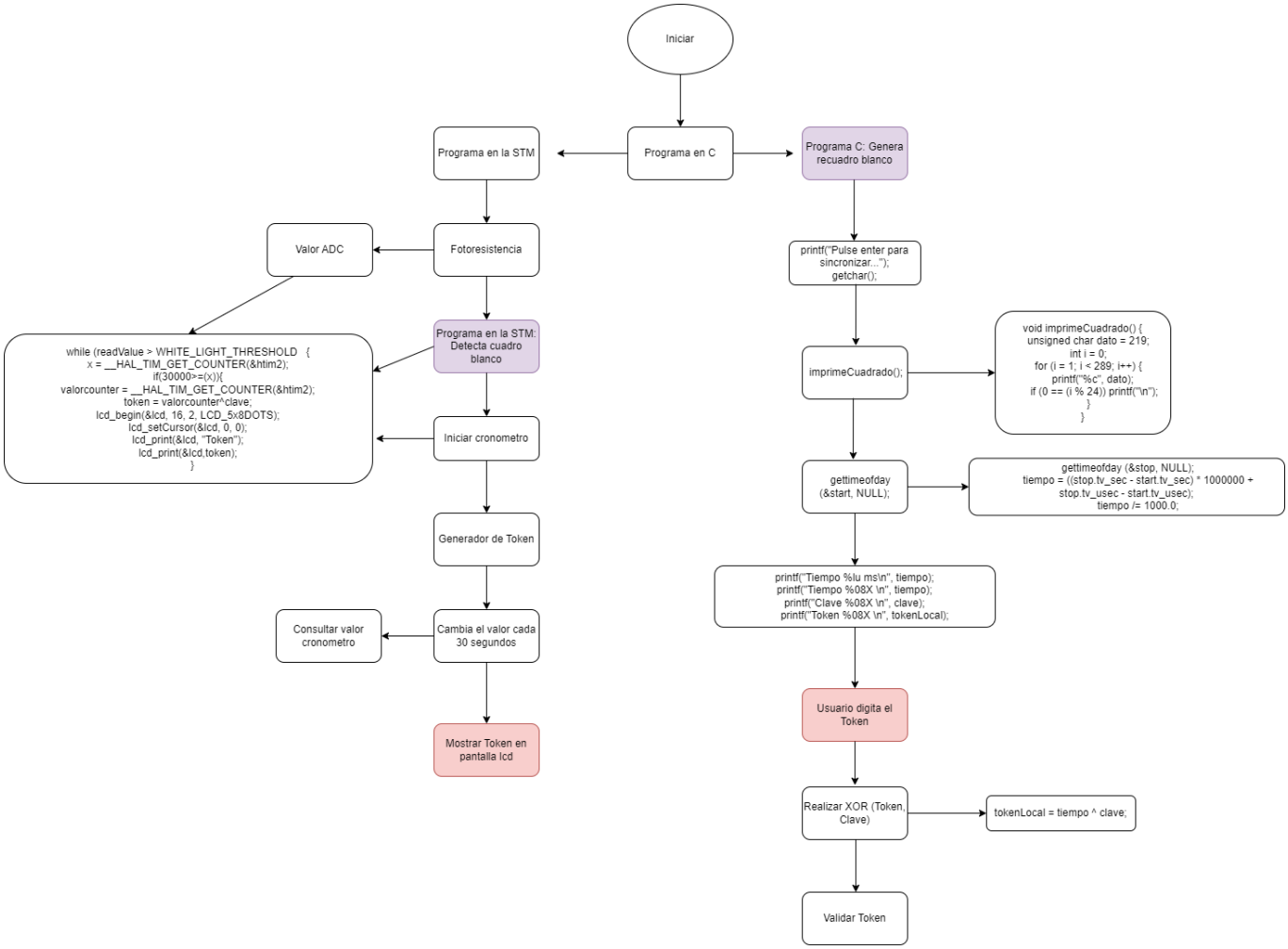

Figura 8. Diagrama de flujo del código

VII. Referencias

[1] “Stm32f411ce.” [Online]. Available:https://www.st.com/en/microcontrollers-microprocessors/stm32f411ce.html

[2] (S/f). Fastercapital.com. Recuperado el 4 de marzo de 2024, de https://fastercapital.com/es/contenido/Desentranando-el-papel-de-la-operacion-XOR-en-el-calculo-CRC.html#Comprensi-n-del-papel-de-XOR-en-el-c-lculo-de-CRC

[3]Donohue, B. (2014, abril 10). ¿Qué Es Un Hash Y Cómo Funciona? Kaspersky. https://latam.kaspersky.com/blog/que-es-un-hash-y-como-funciona/2806/

[4]Fotoresistencia. (s/f). Edu.co. Recuperado el 4 de marzo de 2024, de https://www.unipamplona.edu.co/unipamplona/portalIG/home_74/recursos/visual-basic-para-excel/17052017/u5_fotoresistencia.jsp

[5]Fotoresistencia Resistencia Fotosensible LDR Sensor De Luz. (s/f). MechatronicStore. Recuperado el 5 de marzo de 2024, de https://www.mechatronicstore.cl/fotoresistencia-ldr-5mm/