Explicación SHA256.cpp

Constante K [64]: Esto son valores constantes predefinidos utilizados en cada ronda del proceso de hash. Estos nueros son derivados de operar los primeros 6 números primos.

Función init(): Establece los ocho registros de 32 bits que se utilizan en el cálculo del hash. Estos valores iniciales son específicos del algoritmo SHA-256.

Función rightRotate(): Realiza una rotación circular hacia la derecha de un valor de 32 bits, esta operación es usada múltiples veces en el algoritmo SHA-256.

Argumentos:

- value (uint32_t): Corresponde al valor a rotar.
- count (unsigned int): corresponde a la cantidad de bit a rotar.

Devuelve el valor rotado de 32 bits que se debe guardar en una variable previamente definida.

Función transform(): Esta es la función principal del procesamiento. Toma un bloque de 512 bits y actualiza el estado interno. Dentro de esta función se realizan múltiples operaciones bit a bit, combinadas con las constantes predefinidas y las rotaciones para producir una salida.

Argumentos:

- `data` (const uint8_t [64]): Un bloque de 64 bytes de datos que se va a procesar.

No tiene valor de retorno porque modifica directamente el estado interno.

Función update(): Esta función procesa los datos de entrada en bloques de 64 bytes. Si el tamaño de los datos no es múltiplo de 64, el bloque final se rellena y se añade el padding. Luego, llama a `transform()` para procesar cada bloque.

Argumentos:

- `data` (const uint8_t*): Un puntero al bloque de datos de entrada.
- `len` (uint32_t): La longitud del bloque de datos de entrada en bytes.

Sin valor de retorno.

Función final(): Toma el estado interno final después de procesar todos los bloques y convierte cada registro de 32 bits en 4 bytes para formar el hash final de 256 bits.

Argumentos:

- `hash` (uint8_t [32]): Un arreglo donde se almacenará el hash resultante de 256 bits (32 bytes).

No tiene valor de retorno porque llena el arreglo hash con el valor final.

Función hash(): Esta función combina las llamadas a `init()`, `update()` y `final()` para generar el hash SHA-256 de un texto de entrada. Es la función que se utilizara para obtener el hash de un mensaje.

Argumentos:

- `text` (const char*): El texto de entrada que se va a cifrar.
- `hash` (uint8_t [32]): Un arreglo donde se almacenará el hash resultante.

Sin valor de retorno debido a que el hash resultante se almacena en el arreglo hash.

Explicación BasicUsageSHA_256.ino

Este programa es un ejemplo simple de cómo utilizar la librería SHA256 para calcular y mostrar el hash SHA-256 de una cadena de texto en placas. La salida en el monitor serial será el hash en formato hexadecimal, que representa la firma única del texto `"APVCMSJ2024"`.