

Design and Implementation of a Simple Password-Protected Safe Using Arduino

Misael Gallardo
Universidad de Atacama
Ingeniería Civil Informática
Copiapó, Chile
Misael.Gallardo.19@alumnos.uda.cl

Celeste Sarmiento
Universidad de Atacama
Ingeniería Civil Informática
Copiapó, Chile
Celeste.Marambio.21@alumnos.uda.cl

Abstract—This project focuses on the design and implementation of a basic sequential system for a password-protected safe, utilizing an Arduino microcontroller and two buttons as input devices. Modeled as a Mealy finite-state machine, the system allows users to enter a sequence of button presses to serve as the password, granting access for a limited time upon the correct input. If an incorrect input is detected at any point, the system generates an error signal before resetting.

The primary goal of the project is to provide a functional example of an embedded system that interacts with hardware inputs and offers visual feedback through LEDs. Using Arduino code, the system processes button inputs, controls LED outputs, and verifies the password sequence. A thorough documentation and modeling approach is taken to ensure that each development step is clearly defined and well understood, with the aim of reinforcing knowledge of automata theory and finite-state machines. Thus, this project serves not only as a practical application of embedded systems but also as an educational tool for exploring the design and optimization of sequential logic systems. By applying a Mealy machine model, the system was optimized to reduce state complexity while maintaining correct functionality, thus demonstrating the benefits of finite-state machine theory in real-world applications.

Keywords—arduino, embedded systems, microcontroller security, password authentication

I. Metodología

Para asegurar el correcto funcionamiento del sistema, se identificaron los siguientes pasos clave para asegurar el correcto funcionamiento del sistema y permitir que su desarrollo fluyera sin mayores percances:

- 1) **Definición de requisitos:** En esta etapa, se estableció la necesidad de utilizar dos botones como entradas, una secuencia de cuatro pasos como contraseña, y LEDs como indicadores de estado. Se decidió que el sistema debería reiniciarse si la secuencia era ingresada incorrectamente, proporcionando una mayor seguridad.
- 2) **Diseño del sistema:** Se diseñó el sistema en torno a una lógica secuencial, modelada y optimizada como un autómata de Mealy para reducir la cantidad de estados. Esto garantizó que las entradas fueran procesadas en el orden correcto y que cualquier error en la secuencia reinicia el sistema.
- 3) **Desarrollo del código en Arduino:** Se escribió el código para controlar el flujo de entrada, verificar la secuencia, y proporcionar retroalimentación visual mediante LEDs. Se optimizó la lógica para que el sistema funcionara de manera eficiente, asegurando una respuesta rápida a las entradas.
- 4) **Validación y pruebas:** La realización de pruebas físicas para validar el correcto funcionamiento del sistema es vital, asegurando que el sistema responda apropiadamente tanto a las entradas correctas como a las incorrectas. Además, se verifica que el tiempo de activación y el restablecimiento automático funcionaran de acuerdo a lo previsto.

Los principales componentes del sistema son tres:

- 1) **Microcontrolador Arduino Uno**, que actúa como la unidad de procesamiento principal,

gestionando las entradas y salidas del sistema.

- 2) **Dos botones (“A” y “B”)**, utilizados para ingresar la secuencia que será comparada con la contraseña almacenada.
- 3) **Diodos emisores de luz (LEDs)**, que sirven como dispositivos de salida para proporcionar retroalimentación visual al usuario sobre el estado del sistema.

Estos elementos, en conjunto, son los bloques fundamentales con los cuales se puede construir el sistema y la interacción con la caja fuerte como se espera.

II. Diseño del Sistema

El sistema está diseñado en torno a una lógica secuencial, donde el orden de las entradas juega un rol crucial para determinar las salidas. Se trata de un sistema de protección mediante contraseña, en el cual se debe realizar una serie de acciones en una secuencia específica para permitir el acceso. Para permitir un mejor entendimiento del sistema a desarrollar, es posible moldearlo como un autómata de estados finitos, particularmente como un autómata de Mealy, ya que las salidas dependen directamente de las transiciones entre estados y las entradas actuales, lo cual permite una respuesta más rápida y eficiente[1].

En este caso, la combinación de dos botones (“A” y “B”) sigue una secuencia predeterminada (A, A, B, A), la cual debe ingresar correctamente para otorgar acceso. Cualquier desviación respecto a la secuencia establecida reinicia el sistema y proporciona una retroalimentación negativa al usuario mediante los LEDs. Esta retroalimentación es crucial para alertar al usuario sobre el fallo, facilitando el reinicio del proceso sin confusión. Por el contrario, si la secuencia es ingresada correctamente, el sistema activa una señal de alta energía (representada por el encendido del LED) durante 15 segundos, indicando que el acceso ha sido concedido con éxito. Con esta configuración, se identifican en primera instancia cinco estados principales, cada uno con transiciones bien definidas para las combinaciones de entradas (00, 01, 10):

- 1) **Estado inicial (S_0):** El sistema espera el primer input correcto. Cualquier entrada errónea reinicia el sistema. Los LEDs están apagados. Pulsar “B” o no pulsar nada deja al sistema en este estado.
- 2) **Estado intermedio 1 (S_1):** Después de ingresar la primera “A” correctamente, el sistema avanza al siguiente estado. El LED verde parpadea brevemente. Pulsar “B” en este estado causará un retorno a S_0 .
- 3) **Estado intermedio 2 (S_2):** Luego de ingresar la segunda “A”, el sistema avanza al siguiente paso. El LED verde parpadea por un corto periodo. Pulsar “A” provoca un retorno al estado S_0 .
- 4) **Estado intermedio 3 (S_3):** Después de ingresar la “B” correctamente, el sistema se encuentra en el último estado antes de la validación final. El LED verde parpadea un instante. Pulsar “A” implica volver al estado S_0 .
- 5) **Estado final (S_4):** La secuencia ha sido completada correctamente tras ingresar la última “A”. El LED verde se enciende de forma continua durante 15 segundos, indicando que el acceso ha sido concedido.

Cada uno de estos estados está estrictamente definido, lo que permite un control preciso del flujo del sistema. Este diseño garantiza que cualquier error en la secuencia sea detectado de inmediato y que el sistema responda de manera consistente, proporcionando seguridad adicional al proceso.

La figura 1 presenta cada secuencia definida dentro del autómata de Mealy, donde las entradas de los botones (“A” y “B”) y la salida del sistema (“Z”) son representados en el formato “AB/Z”. i.e. Si se presiona “A” mostrará como entrada “10/”, y dará como resultado una salida “1” si pertenece a la secuencia y “0” si es errónea, provocando que vuelva a S_0 . Esto mismo ocurrirá para “B” con la diferencia que el formato será “01/”. En caso que ningún botón sea presionado “00/” la salida será “0”, representando al sistema en un estado de espera de una entrada válida. En la tabla 1 puede apreciarse cómo las entradas dado un estado, da lugar al siguiente estado dada las entradas estándar (00, 01, 10).

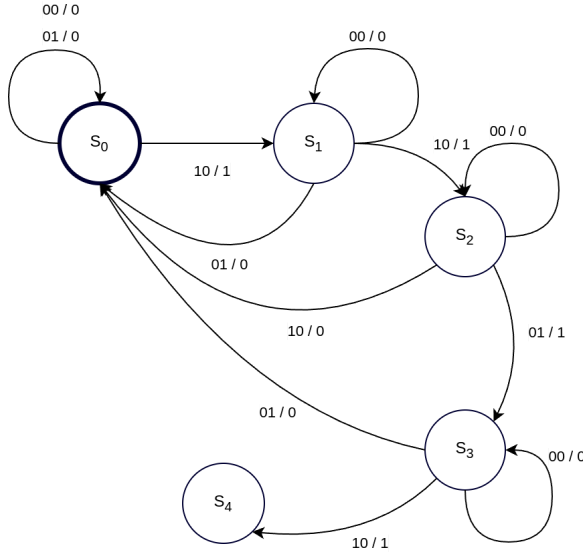


Fig. 1. Diagrama no optimizado de una Máquina de Mealy para una caja fuerte protegida por contraseña.

TABLA I. Tabla de Estados Finitos.

Estado Actual	A	B	Estado Siguiente	Z
0	0	0	0	0
0	0	1	0	0
0	1	0	1	1
1	0	0	1	0
1	0	1	0	0
1	1	0	2	1
2	0	0	2	0
2	0	1	3	1
2	1	0	0	0
3	0	0	3	0
3	0	1	0	0
3	1	0	4	1
4	0	0	—	1
4	0	1	—	1
4	1	0	—	1

El autómata de Mealy es optimizable al observar que el estado S_4 no realiza nada sobre sí mismo; es un estado transitorio el cual no interactúa con las entradas estándar. Lo que nos interesa de este estado es la salida (Z) que provoca un nivel alto de energía, el cual permite que la caja fuerte se abra durante unos 15 segundos. Dado que este comportamiento puede ser gestionado en los estados anteriores, podemos eliminar S_4 para simplificar el diseño.

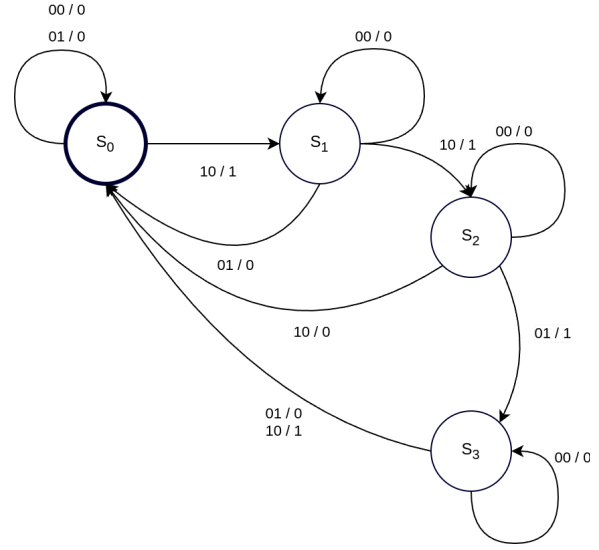


Fig. 2. Nuevo autómata de Mealy, el estado S_4 ha sido eliminado, ahora la función de salida es asociada directamente con la transición de S_3 , permitiendo una respuesta más inmediata.

Al optimizar el diagrama, se han redistribuido las transiciones de salida (1), asociando la activación de la caja fuerte con las transiciones finales del estado S_3 . Esto significa que al ingresar la secuencia correcta (A, A, B, A), no es necesario pasar por un estado adicional (como S_4) para generar la salida deseada, eliminando redundancias.

III. Implementación

La implementación técnica de este proyecto se centró tanto en el desarrollo del código como en la integración de los componentes físicos, manteniendo la simplicidad del hardware gracias a la optimización de la lógica mediante el autómata de Mealy de la figura 3. La eliminación de un estado innecesario permite una implementación más eficiente.

El microcontrolador Arduino Uno fue programado para manejar las entradas de los dos botones y procesarlas como una secuencia secuencial. La codificación se dividió en las siguientes fases:

- 1) **Inicialización del sistema:** Se definen las variables necesarias para los botones (A y B) y los tres LEDs utilizados para el control. Se configuran los pines de entrada y salida del microcontrolador, y se inicializan los contadores y variables necesarias para gestionar la secuencia.

- 2) **Lectura de entradas:** El usuario presiona uno de los botones, el Arduino registra la entrada y la compara con la contraseña predefinida (A, A, B, A). En esta etapa, el código está estructurado de forma que cada estado se gestiona mediante el autómata, procesando la secuencia según los valores de entrada.
- 3) **Comparación de la secuencia:** Después de cada entrada, el sistema compara el valor ingresado con la secuencia correcta. Si el valor es incorrecto, el sistema reinicia el contador de la secuencia y los LEDs parpadean para notificar al usuario del error. Al utilizar el modelo de Mealy, la salida (indicada por los LEDs) se ajusta inmediatamente al momento en que la entrada es procesada.
- 4) **Acceso concedido:** Si se ingresa la secuencia correcta, el Arduino envía una señal de activación que enciende un LED durante 15 segundos, indicando que el acceso ha sido concedido. El estado final se elimina gracias a la optimización del autómata, y la salida de activación ocurre al completar la secuencia sin necesidad de estados intermedios adicionales.
- 5) **Restablecimiento automático:** Después del periodo de 15 segundos, el sistema se reinicia automáticamente, listo para recibir una nueva secuencia. Este reinicio incluye el restablecimiento del contador y la desactivación del LED, devolviendo el sistema a su estado inicial.

Con ello, el sistema debería ser capaz de manejar todos los estados disponibles del sistema, impidiendo estados no válidos y permitiendo una interacción con el usuario fluida, donde se comunica de manera efectiva los estados y funcionamiento de la máquina.

IV. Coste de Producción

Dado que parte del objetivo de este proyecto, es producir físicamente el sistema, se detallan los costos de los componentes utilizados para el desarrollo, expresados en pesos chilenos:

TABLA II. Precios de los Componentes

N°	Componentes necesarios para producir		
	Componente	Cantidad	Coste Unitario
1	Arduino Uno	1	\$3.980
2	Líneas Dupont M+M	15	\$600
3	Breadboard 830 puntos	1	\$1.690
4	LEDs colores	3	\$867
5	Resistencia 1K Ω	3	\$900
6	Resistencia 100K Ω	2	\$1.676
7	Botones de 4 pines	2	\$168
Total			\$9.881

La tabla 2 que se presenta muestra los costos de los componentes necesarios para el sistema. Sin embargo, es importante señalar que los precios que se indican se obtuvieron de mercados chinos en los que se utilizaron modelos no oficiales, como unidades Arduino Uno sin marca. Estos precios tienden a ser más bajos que los de los modelos oficiales y no tienen en cuenta los gastos adicionales, como la mano de obra para el montaje, los derechos de importación y los costes de envío. Esto podría incurrir en un coste total considerablemente mayor.

Este análisis sirve como estimación del coste mínimo en circunstancias ideales, en lugar de ser un reflejo exhaustivo del coste real de fabricación e implementación del sistema.

V. Resultados

El proyecto consiguió el resultado deseado, cumpliendo con todos los requisitos establecidos en un plazo óptimo. La demostración práctica y la implementación se completaron con éxito y el sistema funcionó como se esperaba. La caja fuerte protegida con contraseña funcionó correctamente, brindando información visual a través de LEDs y permitiendo el acceso al presionar la secuencia correcta de botones.

Sin embargo, un problema notable emergió durante la etapa de pruebas. Dado el uso de funciones de espera para controlar el parpadeo de los LEDs, el sistema ocasionalmente es incapaz de registrar la secuencia de botones si el usuario la ingresa

demasiado rápido. Esto requiere por parte de los usuarios introducir la secuencia con al menos medio segundo de espera entre cada pulsación de botones para asegurar una correcta detección de la secuencia de botones. Esta limitación deja la ventana abierta para mejorar y optimizar el diseño actual de la caja fuerte, con el objetivo de prevenir las pérdidas en la lectura de las entradas.

A pesar de este inconveniente, el comportamiento del sistema bajo condiciones normales es estable y confiable. El problema previamente mencionado otorga una valiosa perspectiva sobre el manejo del tiempo sobre los sistemas embebidos. Eso remarca la necesidad de, en futuras iteraciones, diseñar e implementar sistema de detección de entradas más robustos, posiblemente haciendo uso de mecanismos basados en interrupciones, o técnicas de rebote, de tal manera que se asegure una respuesta acertada a la lectura de entradas, independiente de la velocidad a la que se ingresen.

VI. Conclusión

En conclusión, mientras el sistema funciona como se esperaba bajo un uso típico, la limitación observada sirve como una experiencia importante. Enfatiza la importancia de balancear los tiempos de espera del sistema que tienen como propósito retroalimentar a los usuarios.

El proyecto de la caja fuerte a pesar de que se terminó de una manera rápida durante el taller, el análisis fue algo que estuvo trabajando una semana entera; Que si el cable va así; si pongo las resistencia de esta manera, ¿Ocurrirá esto? Incluso la implementación del código acabó siendo lo más complicado de lo esperado, debido a que al principio no era posible hacer que la caja fuerte abriera correctamente por errores en los bucles y en las restricciones necesarias.

Aun así, este fue un proyecto divertido, un desarrollo didáctico que permite comprender mucho más allá el funcionamiento de los sistemas embebidos, y aprender que el Arduino funciona de una manera concreta, y si no es así pues o no funciona, o crea falsos positivos dolorosos de corregir.

VII. Referencias

- [1] M. Melendez, “ARQUITECTURA DE SISTEMAS DIGITALES DE CONTROL: Un Enfoque Práctico”, pp. 6-11, no publicado.