



INFORME DE LABORATORIO 4:

Gestión de Inventario con RFID

Autores: *Santiago Giraldo Tabares, Ana María Velasco
Montenegro*

*Laboratorio de Electrónica Digital 3
Departamento de Ingeniería Electrónica y de Telecomunicaciones
Universidad de Antioquia*

Resumen

En este documento se desarrolla un sistema de gestión de inventario utilizando la tecnología RFID, enfocado en una bodega que maneja cinco productos distintos. El sistema integra elementos de hardware como el lector RFID, teclado matricial y display, y se programa utilizando técnicas de polling e interrupciones. El sistema permite realizar transacciones de entrada y salida de productos, manteniendo un registro actualizado incluso en casos de interrupción de energía. Se destacan algunas características del sistema como la autenticación de usuarios, control de la capacidad de almacenamiento de la bodega y visualización en tiempo real del estado del inventario. Se documenta el proceso de diseño, desarrollo y pruebas, usando la Raspberry Pi Pico, y se discute la efectividad del sistema en términos de manejo de inventario.

Palabras clave: *Raspberry Pi Pico, teclado matricial, RFID, display*

Introducción

La necesidad de una gestión eficiente del inventario en la industria agrícola es cada vez más importante en un mundo donde la optimización de recursos y la sostenibilidad son clave. En esta práctica combina la el software y el hardware por medio de una aplicación real de la industria mediante sistemas embebidos. Se enfoca en el desarrollo de un sistema de gestión de inventario utilizando la tecnología de identificación por radiofrecuencia (RFID) y la Raspberry Pi Pico, destacando la relevancia de estas tecnologías en la automatización y el control de procesos.

El propósito es diseñar e implementar un sistema que no solo mejore la eficacia en el manejo de inventarios para una bodega de insumos agrícolas, sino que también incorpore medidas de seguridad robustas para garantizar que solo el personal autorizado pueda gestionar y acceder al sistema. Utilizando la Raspberry Pi Pico, una plataforma versátil y económica, el sistema implementa un conjunto de funcionalidades críticas como la autenticación de usuarios mediante TAGs RFID y contraseñas, y la capacidad de realizar transacciones de entrada y salida de productos, manteniendo un registro preciso y actualizado del inventario.

Este documento detalla el proceso desde el concepto inicial hasta la implementación final del sistema, describiendo las diversas etapas de diseño del hardware y del

software. Se muestra cómo los componentes como los lectores RFID, teclados matriciales, y displays de caracteres se integran en la Raspberry Pi Pico para crear un sistema funcional. Además de la gestión del almacenamiento permanente para preservar los datos ante cortes de energía y la optimización de la interfaz de usuario para facilitar la operatividad en entornos reales.

Marco teórico

Los sistemas embebidos desempeñan un papel muy importante debido a que son dispositivos que realizan tareas específicas de forma autónoma y eficiente. Estos sistemas, combinan hardware dedicado con software específicamente diseñado, se encuentran desde dispositivos móviles y electrodomésticos hasta automóviles y sistemas de transporte avanzados.

0.0.1. Sistemas Embebidos:

Los sistemas embebidos son sistemas computacionales diseñados específicamente para realizar funciones dedicadas dentro de sistemas más grandes, típicamente en tiempo real. Estos sistemas se diferencian de las computadoras de propósito general por estar optimizados para reducir el tamaño, recursos, consumo de energía y costo. Operan bajo restricciones de hardware y software, y a menudo requieren sistemas operativos en tiempo real para manejar eficientemente múltiples tareas y procesos. Los sistemas embebidos son esenciales en aplicaciones que van desde dispositivos domésticos hasta controles industriales complejos, y son críticos para implementar interfaces de usuario avanzadas, controlar operaciones automatizadas y asegurar la comunicación entre dispositivos.

0.0.2. Microcontroladores y la Raspberry Pi Pico:

La Raspberry Pi Pico es un ejemplo destacado de microcontrolador. Basada en el RP2040, ofrece un núcleo dual ARM Cortex-M0+ a 133 MHz, con 264 KB de RAM y 2 MB de almacenamiento en flash, proporcionando un equilibrio entre capacidad de procesamiento y eficiencia energética. Su compatibilidad con entradas analógicas y digitales, junto con interfaces seriales como UART, SPI e I2C, permite una vasta gama de aplicaciones. La Pico puede ser programada en C/C++ o Python, lo que la hace accesible para principiantes, mientras que su bajo

costo la hace ideal para proyectos de volumen y educativos.

0.0.3. Tecnología RFID:

La RFID es una tecnología que utiliza campos electromagnéticos para transmitir datos entre un lector y un objeto etiquetado para identificación y rastreo. Las etiquetas pueden ser activas, pasivas o semi-pasivas, cada una con diferentes aplicaciones y limitaciones. Las etiquetas pasivas, las más utilizadas en la gestión de inventarios, no requieren una fuente de energía propia, siendo activadas por las ondas de radio emitidas por los lectores. La capacidad de las etiquetas RFID para ser leídas sin contacto directo ni línea de visión directa facilita el seguimiento eficiente y automático de grandes cantidades de productos o activos.

0.0.4. Protocolos de Comunicación: Serial, I2C, SPI

Los protocolos de comunicación son fundamentales para integrar diversos dispositivos electrónicos. SPI es un protocolo basado en un sistema maestro-esclavo, ideal para la transmisión de datos a alta velocidad sobre cortas distancias. I2C, por otro lado, utiliza un sistema de bus con capacidad para múltiples maestros y esclavos, ideal para conectar múltiples dispositivos de baja velocidad sobre dos cables. Estos protocolos son cruciales en aplicaciones donde múltiples sensores y actuadores deben comunicarse con un microcontrolador central, como en la gestión de inventarios.

0.0.5. Polling e Interrupciones

El polling es un método de entrada/salida que involucra la verificación constante de la condición de un periférico por parte de la CPU. Aunque es simple de implementar, puede llevar a una alta carga de CPU y latencia en la respuesta a eventos. Las interrupciones, en contraste, son señales emitidas por dispositivos periféricos hacia la CPU, indicando que un evento importante ha ocurrido, lo que permite que la CPU responda de manera inmediata, mejorando la eficiencia del sistema al reducir la carga de CPU y la latencia.

0.0.6. Almacenamiento permanente

El almacenamiento de datos de forma permanente es crucial en la gestión de inventarios para asegurar la conti-

nidad del negocio y la integridad de los datos. Las soluciones de almacenamiento permanente en sistemas embebidos incluyen EEPROM y memorias flash, que pueden retener datos incluso cuando el dispositivo está apagado. Estas tecnologías son esenciales para registrar transacciones y cambios en el inventario que deben persistir a través de reinicios y cortes de energía.

0.0.7. Interfaces de usuario para sistemas embebidos

Las interfaces de usuario en sistemas embebidos, como teclados matriciales y displays LCD, son fundamentales para la interacción humana con el sistema. Estos componentes permiten a los usuarios ingresar comandos y recibir retroalimentación del sistema, haciéndolos esenciales en aplicaciones donde la configuración manual del sistema o la visualización de datos es necesaria. El diseño de estas interfaces debe considerar la usabilidad, legibilidad y la eficiencia en la interacción.

Procedimiento:

Para este implementar el sistema de este laboratorio, de la gestión de inventario con RFID en la Raspberry Pi Pico se empezará con la preparación del entorno de desarrollo y la instalación de todas las herramientas y librerías necesarias para programar y debuggear la Pico. Se realizan pruebas de los componentes con Arduino para garantizar su funcionalidad y comprender parte de la teoría. Se procede a escribir en cada tag la información solicitada por medio de la aplicación "MI FARE Classic Tool". Se procede a configurar el hardware conectando el módulo RFID, el display LCD I2C, y el teclado matricial a la Raspberry Pi Pico de acuerdo a los diagramas de conexión. Cuando esté configurado el hardware, se procede a cargar los programas en la Pico para manejar la lectura de los tags RFID, la entrada de datos a través del teclado, y la visualización de información en el LCD. El sistema se prueba de forma modular para asegurar que todos los componentes funcionan correctamente y que el sistema puede manejar tanto la entrada como la salida de inventario, actualizando y mostrando la información en tiempo real. El sistema deberá ser capaz de mantener la integridad de los datos incluso en caso de fallo de energía, garantizando así la confiabilidad y la seguridad de la gestión de inventario.

0.0.1. Configuración del hardware y entorno de desarrollo:

Involucra la preparación y configuración de la Raspberry Pi Pico junto con todos los componentes necesarios como el teclado matricial, la pantalla LCD, el módulo RFID, los tags y las tarjetas.

0.0.2. Configuración de los tags y tarjetas

Se realiza la asignación de la tarjeta de administrador y de bodega, esto se hace mediante el uid de cada tarjeta para así ser identificadas rápidamente.

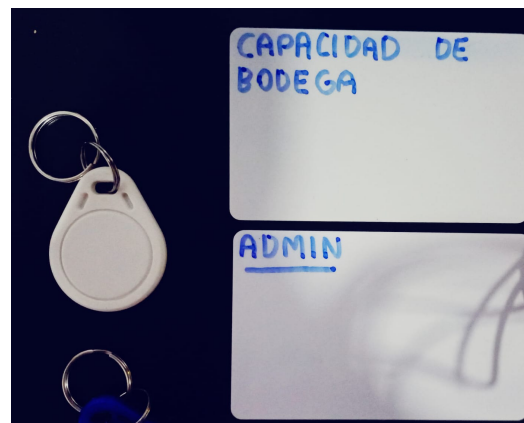


Figura 0-1: Tarjetas admin y bodega

Se procede a configurar cada tag por medio de una aplicación llamada "MI FARE Classic Tool".^{en} donde se escoge un bloque y un sector para escribir en hexadecimal, lo cual se puede observar en la siguiente imagen:

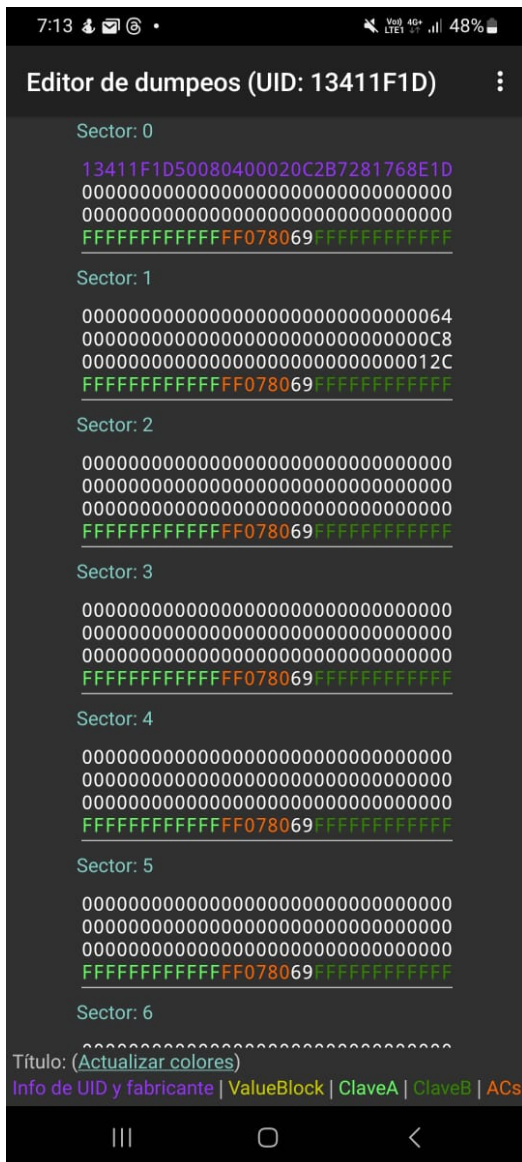


Figura 0-2: Secciones aplicación

Se programa un código el cual permite mostrar para posteriormente extraer esa información almacenada en cada tag, los cuales son: el tipo de producto, número de ítems en la caja y el precio unitario de cada ITEM,

```
15:26:40.907 -> Aproxime la tarjeta al lector...
15:26:42.649 -> Precio Unitario: $300
15:26:42.649 -> Número de ítems: 200
15:26:42.685 -> Tipo de Producto: Pan
```

Figura 0-3: Items tags

0.0.3. Análisis previo:

Se realiza el análisis previo del programa a resolver por medio de un diagrama del flujo el cual permite tener claridad del proceso que se debe realizar. Esto es muy importante porque así se tiene una idea también de lo que se necesita y el momento en que se necesita. Se observa en la siguiente imagen:

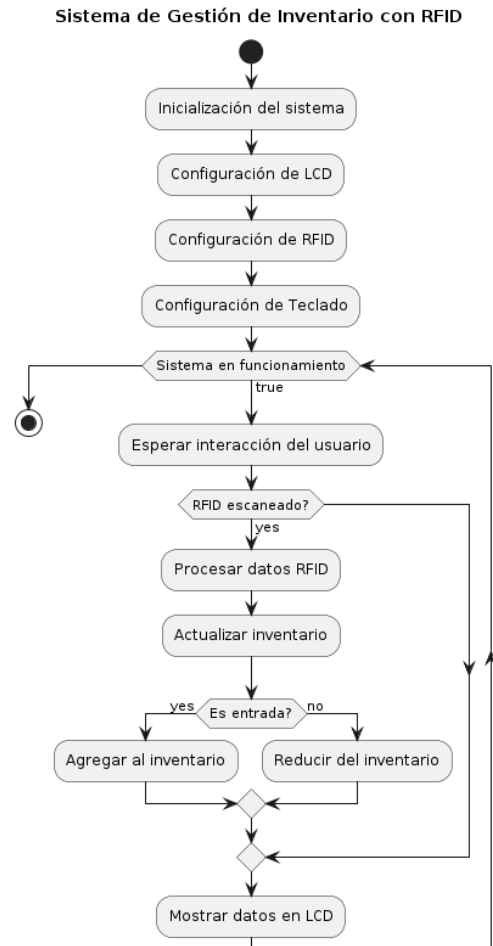


Figura 0-4: Diagrama de flujo general

0.0.4. Módulos a usar:

- **Módulo de inventario** En este módulo se gestiona un sistema de inventario con RFID en una Raspberry Pi Pico, comenzando con la configuración inicial de periféricos como teclados y displays. Facilita la autenticación del usuario mediante RFID y un PIN, seguido por la configuración de la capacidad de la bodega a través de

RFID. Posteriormente, el sistema opera continuamente leyendo y mostrando datos de productos RFID, permitiendo una gestión eficiente y segura del inventario.

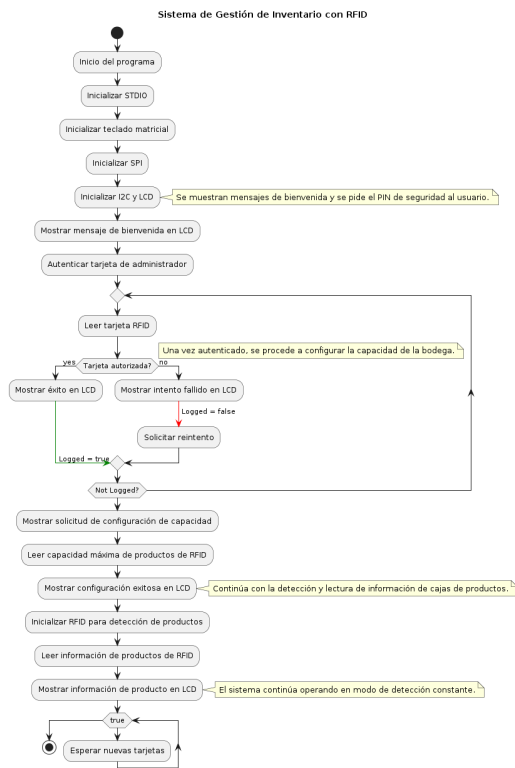


Figura 0-5: Diagrama de flujo inventario

- **Módulo de LCD** En este módulo se inicializa y controla un display LCD mediante I2C, mostrando información de manera cíclica y continua. A continuación se presenta un diagrama ilustrativo:

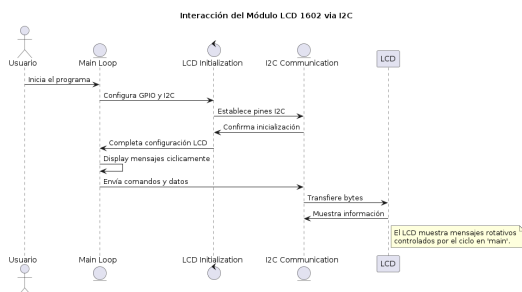


Figura 0-6: Diagrama de flujo LCD

- **Módulo MRFC522** En este módulo RFID

MRFC522 en un Raspberry Pi Pico. Se debe tener en cuenta la inicialización del módulo, autenticación, lectura y escritura de tarjetas RFID, así como manejo de la antena y comandos específicos para tarjetas MIFARE. Utiliza la librería hardware/spi para la comunicación SPI. También contiene definiciones detalladas para los registros y comandos del módulo MRFC522, facilitando la manipulación a bajo nivel del dispositivo RFID.

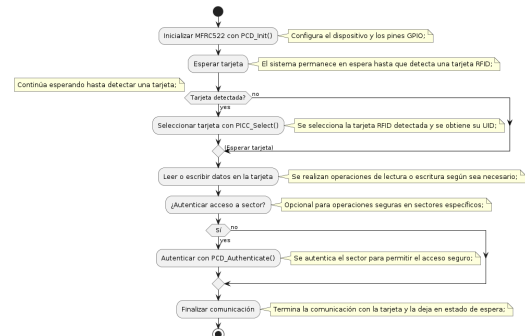


Figura 0-7: Diagrama de flujo Modulo MRFC522

- **Módulo Keypad** En este módulo se inicializan los pines GPIO, se maneja la detección de pulsaciones de teclas con debouncing para evitar lecturas erróneas, y se permite habilitar interrupciones para reaccionar a las pulsaciones. Se utiliza máscaras de bits para identificar eficientemente la tecla presionada basándose en la configuración de filas y columnas.

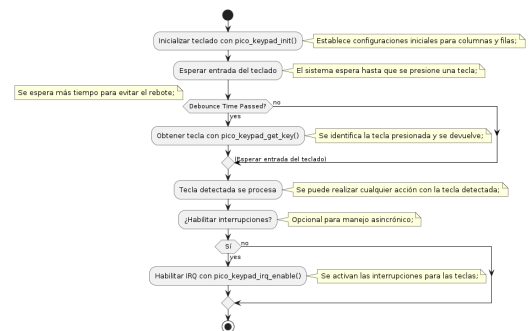


Figura 0-8: Diagrama de flujo Modulo Keypad

0.0.5. Circuito implementado:

Se puede observar el circuito completo implementado con todos los elementos como el módulo MRFC522,

pantalla LCD, tarjetas, tags, teclado matricial y la Raspberry Pi Pico.

A continuación se mostrarán cada uno de los elementos del circuito:

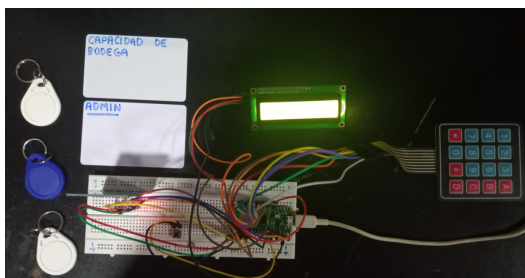


Figura 0-9: Circuito con todos los componentes

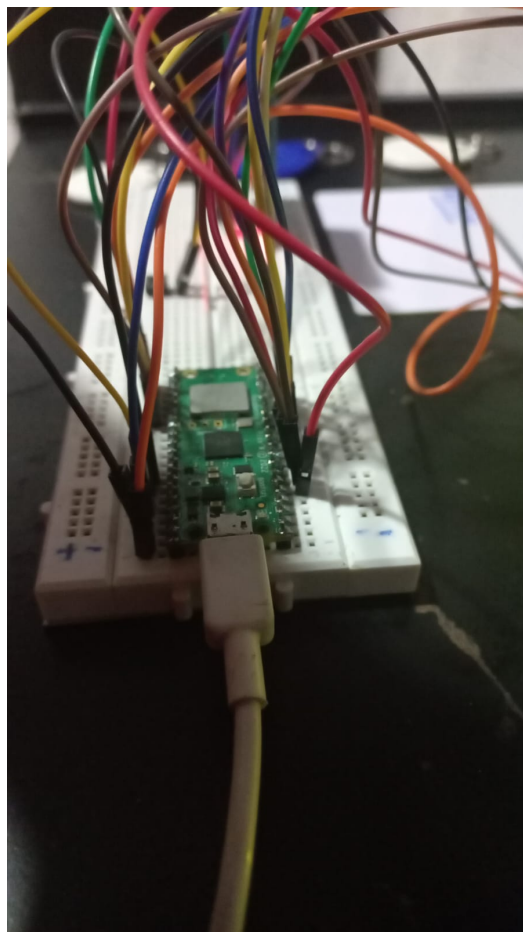


Figura 0-11: Circuito Raspberry Pi Pico

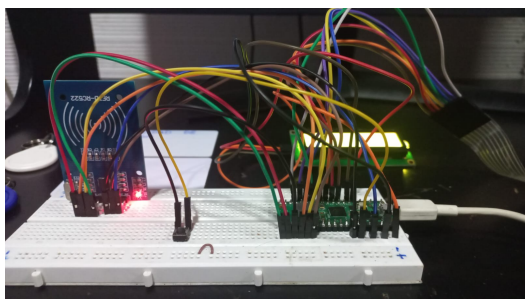


Figura 0-10: Circuito con todos los componentes

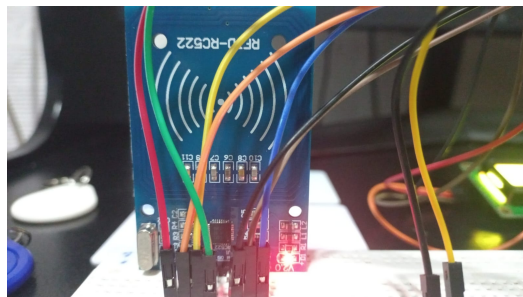


Figura 0-12: Circuito RFID

que aumenta la precisión en la detección de pulsaciones. Cada vez que una tecla es presionada, el código comprueba la correspondencia entre filas y columnas para determinar cuál tecla fue activada. También permite configurar interrupciones para gestionar eventos de pulsación, haciendo que el sistema sea más interactivo y eficiente.

```
char pico_keypad_get_key(void) {
    int row;
    uint32_t cols;

    cols = gpio_get_all();
    cols = cols & all_columns_mask;

    if (cols == 0x0) {
        return 0;
    }

    for (int j = 0; j < 4; j++) {
        gpio_put(_rows[j], 0);
    }

    for (row = 0; row < 4; row++) {
        gpio_put(_rows[row], 1);

        busy_wait_us(10000);

        cols = gpio_get_all();
        gpio_put(_rows[row], 0);
        cols = cols & all_columns_mask;
        if (cols != 0x0) {
            break;
        }
    }
}
```

Figura 0-16: Código Keypad

Este código controla una pantalla LCD 1602 a través de una conexión I2C. Se inicializan los pines GPIO usados para I2C y se configura la comunicación I2C con el panel LCD. El programa envía comandos y datos al LCD para mostrar texto. Los comandos incluyen instrucciones para limpiar la pantalla, posicionar el cursor, y controlar el modo de entrada y visualización del LCD. Se utiliza una técnica de envío de bytes en dos partes (nibbles) para la comunicación, y se incluye un mecanismo para "toggle" del pin de habilitación del LCD que asegura que los comandos y datos sean procesados adecuadamente. El bucle principal del programa muestra mensajes alternativos en la pantalla, demostrando la capacidad del LCD para exhibir texto dinámico.

```
// Modes for lcd_send_byte
#define LCD_CHARACTER 1
#define LCD_COMMAND 0

#define MAX_LINES 2
#define MAX_CHARS 16

/* Quick helper function for single byte transfers */
void i2c_write_byte(uint8_t val) {
#ifdef i2c_default
    i2c_write_blocking(i2c_default, addr, &val, 1, false);
#endif
}

void lcd_toggle_enable(uint8_t val) {
    // Toggle enable pin on LCD display
    // We cannot do this too quickly or things don't work
#define DELAY_US 600
    sleep_us(DELAY_US);
    i2c_write_byte(val | LCD_ENABLE_BIT);
    sleep_us(DELAY_US);
    i2c_write_byte(val & ~LCD_ENABLE_BIT);
    sleep_us(DELAY_US);
}
```

Figura 0-17: Código LCD

- **Código módulo RFID MFRC522** El código del módulo RFID MFRC522 configura y maneja un lector de tarjetas RFID para interactuar con tarjetas utilizando la Raspberry Pi Pico. Este código establece la comunicación a través del protocolo SPI, inicializa el dispositivo y realiza ajustes en los registros para operar correctamente. Permite autenticar tarjetas y realizar operaciones de lectura y escritura en ellas. Además, ofrece funcionalidades para ajustar la potencia del lector, garantizar la seguridad en las operaciones con tarjetas y ejecutar comandos específicos del protocolo MIFARE.
- **Código módulo LCD**

Obstáculos en la implementación

En el desarrollo del sistema de gestión de inventario, la parte más desafiante fue la implementación de la lectura de los tags RFID. Este proceso resultó complicado principalmente porque requiere un dominio completo de las múltiples funciones que ofrece la biblioteca del módulo MFRC522 para manipular la información de los tags. Entre estas funciones se incluyen la instanciación, inicialización, lectura, y el control de encendido, apagado y gestión general del PCD (Proximity Coupling Device). La correcta implementación de estas operaciones es

crucial para asegurar la eficiencia y precisión en la recuperación de la información almacenada en los tags RFID.

Además, la escritura sobre los tags RFID presentó desafíos significativos. Esta dificultad se superó gracias al uso de una aplicación móvil desarrollada por la misma empresa que fabrica y vende los tags MIFARE utilizados durante la práctica. Esta herramienta proporcionó una interfaz directa y eficaz para manipular los datos de los tags, facilitando significativamente el proceso de escritura en ellos.

Discusión

El manejo del teclado 4x4 resultó ser la tarea más sencilla, ya que había sido previamente abordado en el curso y su uso está bastante refinado. Comparativamente, este periférico es mucho más simple que el lector RFID. Por otro lado, la utilización del LCD tampoco presentó mayores complicaciones, dado que la comunicación mediante I2C es significativamente más fácil de implementar y cablear que la realizada a través de SPI. Además, existe una abundante cantidad de información y trabajos previos sobre este dispositivo, ampliamente usado tanto a nivel profesional como en el ámbito del hobby.

El módulo MFRC522, en cambio, ofrece una amplia gama de funciones y parámetros que lo convierten en un instrumento bastante versátil. No obstante, comprenderlo completamente se convierte en una tarea prolongada y tediosa. Es importante reconocer que un entendimiento insuficiente de un periférico puede hacer que su integración en el código sea, en el mejor de los casos, muy ineficiente y, en el peor de los casos, directamente imposible.

Además, la documentación sobre este periférico específicamente para la Raspberry Pi Pico es limitada, dado que su uso es más frecuente en proyectos con Arduino, lo que representa un desafío adicional.

La implementación final del proyecto, que incluye todos los códigos fuente, documentación, procesos los cuales son necesarios para comprender y replicar el laboratorio número 4 de la materia Electrónica Digital 3 desarrollado para la Raspberry Pi Pico, está disponible públicamente para consulta y descarga en GitHub. Este repositorio representa el trabajo en equipo y las iteraciones de desarrollo que han caracterizado este proyecto desde su inicio. El link al repositorio es el siguiente: [GitHub Inventario con RFID](#)

Recomendaciones

En el desarrollo de sistemas embebidos, seleccionar hardware bien documentado y con soporte comunitario robusto es esencial, ya que facilita la implementación y acelera el aprendizaje. Para periféricos complejos o menos documentados, como el módulo RFID MFRC522 en Raspberry Pi Pico, es crucial invertir tiempo en investigación y comprensión detallada antes de la implementación.

Es importante familiarizarse profundamente con las bibliotecas y explorar recursos de comunidades en línea para soporte adicional. Mantener documentación detallada del desarrollo y adoptar un enfoque incremental en la implementación y prueba de periféricos ayuda a identificar y resolver problemas eficientemente, optimizando el proceso de desarrollo.

Conclusiones

- La implementación de periféricos puede variar enormemente en complejidad, dependiendo en gran medida de la documentación y soporte disponibles. La familiaridad previa del desarrollador con el hardware específico también juega un papel crucial en facilitar o complicar este proceso.
- Los periféricos más complejos, como el módulo RFID MFRC522, presentan desafíos considerables debido a la necesidad de comprender profundamente una amplia gama de funciones y parámetros. La falta de documentación específica para ciertos microcontroladores, como la Raspberry Pi Pico, agrega una capa adicional de dificultad.
- Periféricos como teclados 4x4 y pantallas LCD, que utilizan comunicación I2C, suelen ser más sencillos de implementar y manipular. Esto se debe en parte a la mejor documentación disponible y a la experiencia que se tienen con estos dispositivos.
- La experiencia previa y el acceso a recursos de soporte adecuados son fundamentales para el desarrollo eficiente de sistemas embebidos. Estos factores pueden determinar la rapidez y eficacia con la que se pueden integrar nuevos periféricos en un proyecto.
- La complejidad en la implementación de periféricos no solo afecta la curva de aprendizaje del desarrollador, sino también el tiempo general de desarrollo y la eficiencia del sistema final. Periféricos mal entendidos o incorrectamente implementados

pueden conducir a errores frecuentes y problemas de rendimiento.

- Existe la necesidad de mejorar la documentación y las herramientas de soporte para periféricos menos comunes o más complejos. Esto no solo facilitaría el trabajo sino que también ampliaría las posibilidades de uso de tecnologías emergentes en diversos proyectos.

Referencias

[1] Raspberry Pi Foundation. (2021). Raspberry Pi Pico Documentation. [2] NXP Semiconductors. (2020). MFRC522 Standard performance MIFARE and NTAG frontend. [3] Doe, J. (2022). Integrating MFRC522 RFID Reader with Raspberry Pi Pico. Electronic Hobbyist [4] Smith, A. (2023). Embedded Systems Design Course. University of Electronics Online.