

1^{er} laboratorio: GPIO

Rodrigo Karlen

UTEC

Ingeniería Mecatrónica

Fray bentos, Uruguay

rodrigo.karlen@estudiantes.utec.edu.uy

Francisco Mayol

UTEC

Ingeniería Mecatrónica

Fray bentos, Uruguay

francisco.mayol@estudiantes.utec.edu.uy

Luis Nunez Da Rosa

UTEC

Ingeniería Mecatrónica

Fray bentos, Uruguay

luis.nunezdarosa@estudiantes.utec.edu.uy

I. RESUMEN

En este trabajo se desarrolló un sistema de control completo para una lavadora automatizada y una matriz de LEDs, utilizando un enfoque de programación en lenguaje ensamblador con el microcontrolador ATmega328P. El principal objetivo fue diseñar e implementar un control eficiente que permitiera gestionar de forma precisa los distintos modos de operación de la lavadora, incluyendo el lavado, centrifugado y secado. Se buscó, además, facilitar la interacción en tiempo real mediante comunicación UART, lo que permitió tanto el monitoreo continuo del sistema como su control remoto.

Adicionalmente, se desarrolló un sistema de visualización basado en una matriz de LEDs, programada para mostrar un mensaje desplazable de manera clara y suave. Esta matriz, además, cuenta con la opción de seleccionar entre varios modos de visualización.

II. INTRODUCCIÓN

En el siguiente informe abordaremos dos situaciones problemáticas a resolver mediante código Assembler con el uso de un microcontrolador ATmega328P. Se propone el desarrollo del control de una lavadora automática. El mismo abarca tanto la simulación como la implementación física de los modos de operación de la lavadora (lavado, centrifugado y secado), así como la interacción con un sistema externo mediante comunicación UART. Además, se solicita un sistema de visualización utilizando una matriz de LEDs controlada por el mismo microcontrolador que muestre una frase de al menos 12 caracteres.

III. OBJETIVOS

Objetivo Principal

Desarrollar e implementar una solución para los problemas a resolver, mediante sistemas de control desarrollados a través de Assembler con el Atmega328p.

Objetivos Específicos

- Montar el modelo de lavadora utilizando el kit de Fischertechnik.
- Programar el sistema de control de la lavadora que gestione los modos de operación y permita la comunicación con un sistema externo a través de USART.
- Implementar el control de una matriz de LEDs con mensaje desplazable y selección de modos mediante UART.

IV. MATERIALES

En esta práctica utilizaremos diversos materiales que nos permitirán programar tanto el control de la lavadora en sus diferentes modos de operación como la visualización de un mensaje desplazable en la matriz de LEDs, además de establecer la comunicación UART en ambos sistemas. A continuación, se presenta la lista de materiales necesarios para ambas partes:

- Microcontrolador ATmega328P
- Kit de montaje de Fischertechnik
- Pulsadores
- LEDs indicativos
- Motor de lavadora
- Cables para conexiones eléctricas
- Fuente de alimentación
- Multímetro
- Panel de Iluminación DOLANG

V. MARCO TEÓRICO

ATmega328P

El microcontrolador ATmega328P es un chip de 8 bits basado en la arquitectura AVR, ampliamente utilizado en sistemas embebidos por su versatilidad y eficiencia. Entre sus principales características destacan sus 32 KB de memoria flash, 2 KB de SRAM, y 1 KB de EEPROM, lo que le permite almacenar y ejecutar programas de tamaño moderado. Además, cuenta con 23 pines de entrada/salida digital, 6 de los cuales pueden funcionar como salidas PWM, y 6 canales ADC de 10 bits. También soporta comunicación en serie mediante UART, SPI y I2C. Se adjunta en la Figura 1 una imagen del microcontrolador.

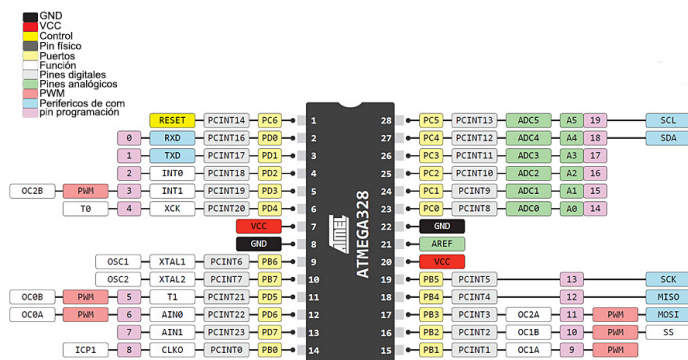


Fig. 1. Microcontrolador Atmega328P

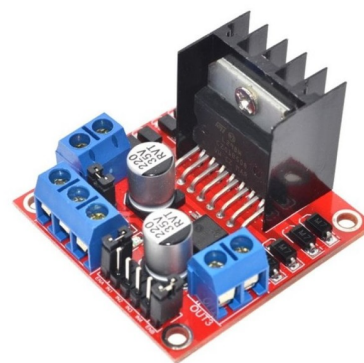


Fig. 2. Controlador de motores L298N

Lenguaje Ensamblador

Por otro lado, el lenguaje ensamblador (assembler) es un lenguaje de bajo nivel que permite el control directo del hardware de un microcontrolador, como el ATmega328P. A diferencia de los lenguajes de alto nivel, el ensamblador proporciona un acceso preciso a los registros y a las instrucciones de la CPU, lo que resulta en programas más eficientes en términos de espacio y tiempo de ejecución. Entre sus ventajas se incluye la capacidad de optimizar el uso de memoria y la ejecución de código en tiempo real, aspectos cruciales para aplicaciones donde los recursos son limitados. Sin embargo, requiere un conocimiento detallado de la arquitectura interna del microcontrolador, ya que cada instrucción se traduce directamente a operaciones de máquina.

Controlador de motores L298N

En la implementación, fue imperativo el uso del controlador de motores L298N, el mismo es un dispositivo que permite manejar motores de corriente continua (DC) y motores paso a paso. Funciona recibiendo señales de control desde un microcontrolador, en este caso del Atmega328p. El L298N utiliza estas señales para regular la dirección y la velocidad de los motores conectados. El L298N puede controlar dos motores a la vez, gracias a sus dos puentes H integrados, que permiten invertir la polaridad de la corriente y así cambiar la dirección de giro. Además, cuenta con pines de entrada para habilitar y deshabilitar los motores, así como para ajustar la velocidad mediante la modulación por ancho de pulso (PWM). Se adjunta en la Figura 2 una imagen del controlador mencionado.

Panel de iluminación DOLANG

A su vez, se debe mencionar el panel de iluminación DOLANG MCUI-03. Es un dispositivo utilizado comúnmente en aplicaciones de entrenamiento y simulación, que integra una matriz de LEDs controlable mediante un microcontrolador. Este panel permite visualizar diferentes patrones o mensajes desplazables, lo que lo hace ideal para la simulación de sistemas de señalización o comunicación visual. El panel se controla a través de señales digitales, y su diseño modular facilita la implementación. Se adjunta en la Figura 3 una imagen del panel.

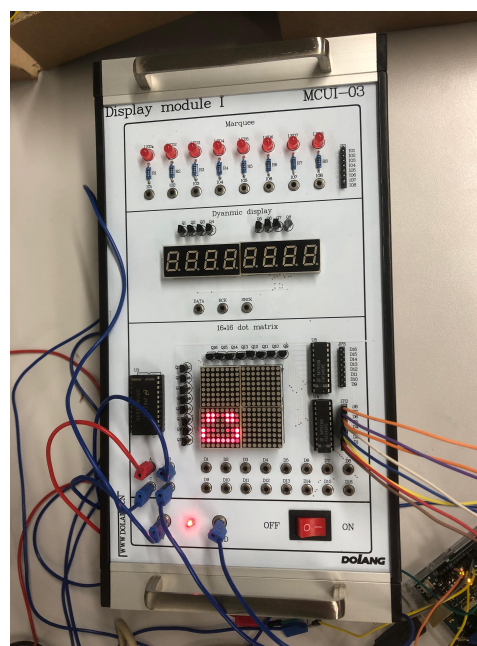


Fig. 3. Panel de iluminación DOLANG MCUI-03

Kit de montaje Fischertechnik

Para albergar y simular el sistema de la lavadora controlada por el Atmega328p, se implementa la lavadora del kit de

montaje de Fischertechnik. El kit de montaje de lavador de Fischertechnik es una herramienta diseñada para simular y estudiar el funcionamiento de una lavadora automatizada. Este kit permite ensamblar un modelo funcional que incluye componentes como motores, sensores y controles, proporcionando una representación simplificada de un sistema de lavado real. El kit está pensado para ser controlado por un microcontrolador, como el ATmega328P, permitiendo el desarrollo de código que gestione los distintos modos de operación de la lavadora, como el lavado, centrifugado y secado. Además, su diseño modular facilita la conexión de periféricos adicionales. Se adjunta en la Figura 4 una imagen del kit mencionado.



Fig. 4. Kit de montaje de Fischertechnik

VI. PROCEDIMIENTO

En esta sección se detallará el procedimiento a seguir para completar con éxito la práctica. El mismo se dividirá en dos partes: la primera estará orientada al montaje del modelo de lavadora utilizando el kit de Fischertechnik, y la segunda abordará dos problemáticas principales: la implementación de sistema de control de la lavadora y la programación de la matriz de LEDs con selección de modos mediante UART.

Parte 1: Montaje del Modelo

En esta sección se detallará el procedimiento para realizar el montaje del modelo de lavadora utilizando el kit de Fischertechnik.

- 1) Leer las primeras páginas del documento *Información básica de los modelos de Fischertechnik* (páginas 2 a 5) para obtener una introducción sobre los componentes del kit y su función.
- 2) Leer las recomendaciones incluidas en el *Manual de montaje para la construcción de modelos*, prestando especial atención a las instrucciones sobre el corte de cables y otras indicaciones importantes para el montaje. En la Figura 5 se presentan las instrucciones básicas del kit del manual de montaje.

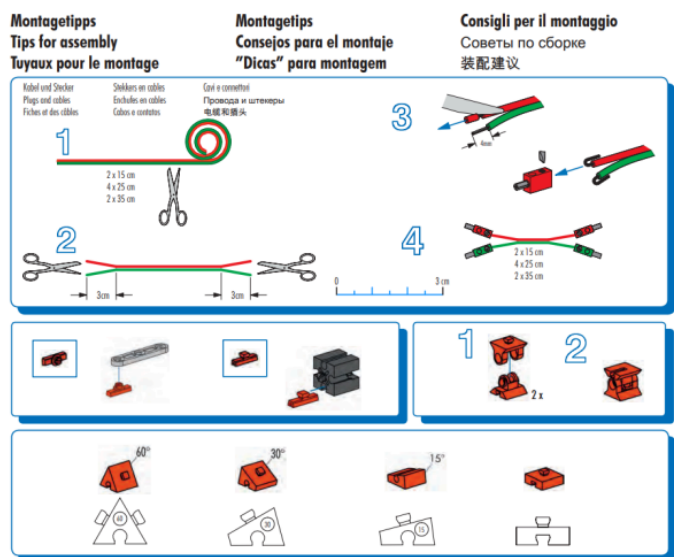


Fig. 5. Instrucciones básicas del kit del manual de montaje

- 3) Ensamblar el modelo de lavadora siguiendo las instrucciones del manual de montaje de Fischertechnik. Durante el ensamblaje, se debe tener en cuenta que el control del modelo será gestionado por el microcontrolador ATmega328P, por lo que no es necesario montar el controlador provisto en el kit de Fischertechnik.

El resultado final de esta sección debe ser un modelo funcional de lavadora, listo para ser controlado por el microcontrolador. En la Figura 6 se muestra cómo debería quedar el modelo de la lavadora, una vez montado siguiendo las instrucciones.

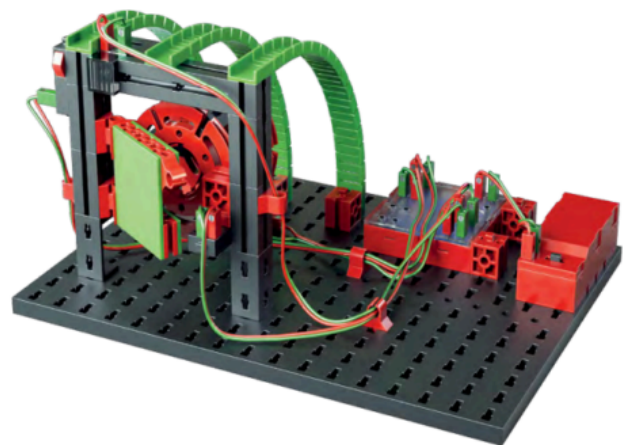


Fig. 6. Modelo de Lavadora ensamblado

Parte 2: Problemáticas

La Parte 2 se dividirá en dos secciones principales: la primera, denominada Problema A, estará enfocada en la implementación del sistema de control para la lavadora, incluyendo

la programación de los diferentes modos de operación y la comunicación mediante USART. La segunda sección, Problema B, se centrará en el control de la matriz de LEDs, donde se programará un mensaje desplazable y se habilitará la selección de modos a través de UART.

Problema A: Lavadora

Párrafo de la problemática: Se está desarrollando una lavadora en fase final de fabricación. Antes de su lanzamiento al mercado, es necesario programar un software que controle su operación, gestione diferentes modos de funcionamiento y permita la comunicación con un sistema de pruebas a través de USART. Esta comunicación es crucial para monitorear el estado de la lavadora durante las pruebas y para iniciar el sistema remotamente.

A continuación se detallan los requisitos que la lavadora debe cumplir para resolver la problemática planteada.

- *Modos de funcionamiento:*

La lavadora cuenta con tres modos principales de funcionamiento, cada uno diseñado para etapas específicas del ciclo de lavado:

- 1) Procedimiento de Lavado: Girar el tambor 2 segundos, hacer una pausa de 1 segundo y repetir este proceso 5 veces. El tiempo de lavado y la cantidad de agua se ajustan según el tipo de carga seleccionada (ligera, media o completa), aumentando 1 segundo por cada carga.
- 2) Procedimiento de Centrifugado: Girar el tambor a máxima velocidad durante 15 segundos. La duración del centrifugado se ajusta según el tipo de carga, aumentando 3 segundos por cada carga.
- 3) Procedimiento de Secado: Girar el tambor hacia la derecha durante 5 segundos, pausar durante 3 segundos, y luego girar hacia la izquierda durante 5 segundos. El número de ciclos de secado varía según el tipo de carga, aumentando 2 segundos por cada carga.

- *Estados de la lavadora:*

La operación de la lavadora se gestiona mediante una serie de estados que definen su comportamiento en cada momento, se debe tener en cuenta el funcionamiento general de una lavadora para implementar los estados necesarios en el sistema.

- La comunicación vía USART permitirá:

- Enviar un mensaje cada vez que la lavadora cambie de estado, permitiendo el monitoreo en tiempo real.
- Iniciar el ciclo de lavado remotamente al recibir la letra "A" desde el sistema externo.

- La lavadora cuenta con LEDs indicativos:

- LED XX: Listo para lavar
- LED XX: Proceso de lavado
- LED XX: Proceso de centrifugado
- LED XX: Proceso de secado
- LED XX: Fin del proceso
- LED XX: Carga ligera

- LED XX: Carga media
- LED XX: Carga pesada

La lavadora también cuenta con los siguientes materiales:

- Un pulsador de inicio
- Un pulsador de selección de carga
- Un sensor (interruptor de seguridad) para poner marcha el lavado cuando la puerta se encuentra cerrada.
- Un sensor de llenado de agua de la lavadora
- El motor de la lavadora

Problema B: Matriz de LEDs

Párrafo de la problemática: Se requiere programar el ATmega328P para controlar la matriz de LEDs, de manera que despliegue un mensaje desplazándose de derecha a izquierda a lo largo de la pantalla conformada por la matriz de leds de DOLANG. El mensaje debe ser legible, claro, y estar compuesto por una frase significativa de al menos 12 caracteres, adicionalmente, el circuito debe conectarse por UART con el cual, al recibir datos del UART, podrá elegir entre el mensaje con desplazamiento o elegir cualquiera de las imágenes ya implementadas en la evaluación continua de matriz de leds.

Requisitos a cumplir:

- 1) El mensaje debe mostrarse de manera visible y desplazarse de forma continua y suave.
- 2) El mensaje mostrado debe tener sentido completo, como "WELCOME HOME" o "HELLO WORLD"
- 3) Implementar una función para ajustar la velocidad del desplazamiento, de modo que el mensaje sea legible en todo momento.
- 4) El código debe estar bien documentado para facilitar futuros ajustes en el contenido del mensaje o en la velocidad de desplazamiento.
- 5) Configurar el circuito para que, mediante la comunicación UART, el usuario pueda seleccionar entre el mensaje con desplazamiento o alguna de las figuras implementadas en la evaluación continua de matriz de LEDs

El sistema debe estar completamente documentado y ser ajustable para facilitar futuras modificaciones tanto del mensaje como de la velocidad de desplazamiento.

VII. RESULTADOS

Aquí se presentan los resultados obtenidos en la práctica, siguiendo detalladamente el procedimiento descrito en la sección correspondiente. Los resultados se dividirán según las diferentes etapas mencionadas en la sección de procedimiento.

Repositorio GitHub

En el siguiente enlace se presenta el repositorio en GitHub, donde se encuentran todos los códigos y las evidencias necesarias solicitadas por la práctica. A medida que se vayan presentando los resultados, se hará referencia a diferentes carpetas y archivos dentro del repositorio. Este contiene diversas versiones de los códigos conforme se fue avanzando en el proyecto, así como imágenes y vídeos que demuestran el funcionamiento de las tareas solicitadas.

Enlace: ***Link a Repositorio GitHub.***

Montaje del Modelo

Se completó con éxito el ensamblaje del modelo de lavadora, sin mayores inconvenientes, siguiendo detalladamente los pasos necesarios para que sea funcional y programable, tal como se describe en la sección de procedimiento. En el repositorio de GitHub (Laboratorios/Laboratorio 1/Imágenes, Videos/Link Video en Drive de demostración de Lavadora.txt), se adjunta un video que demuestra el funcionamiento completo de la lavadora, donde se puede apreciar el modelo completamente ensamblado y operativo.

Problema A: Lavadora

Para el apartado de la problemática A (control de la lavadora), se presenta el diagrama de estados en el repositorio de GitHub (Laboratorios/Laboratorio 1/Documentación/EstadosLavadora.png). Para el flujo de operación y las condiciones de transición entre los estados, se siguieron estrictamente las instrucciones proporcionadas por los modos de funcionamiento, detallados en la sección de procedimiento. Cabe destacar que se decidió incluir solo los estados de las salidas más relevantes en cada estado, dado que, si se representaran todas las salidas del sistema, como los LEDs y otros elementos, se perdería la claridad y el orden del diagrama.

En cuanto al código, se implementó un sistema de seguridad para evitar que el proceso de la lavadora inicie de manera incorrecta o accidental. Para que el sistema esté disponible para realizar un lavado, es necesario que la puerta esté completamente cerrada y que el tanque de la lavadora esté lleno de agua. En ese momento, se enciende un LED que indica que el sistema está listo para iniciar el ciclo de lavado.

Antes de comenzar el proceso, se permite al usuario seleccionar entre tres modos de lavado: ligero, mediano y pesado. Para esta selección, se implementó una interrupción en el programa, la cual incrementa un valor que corresponde a un modo de lavado específico. Este valor tiene un máximo de dos, ya que el valor inicial de cero corresponde al modo ligero.

Una vez seleccionado el modo de lavado deseado, el sistema puede iniciarse mediante un pulsador o a través de la consola utilizando la letra 'A'. El programa recorrerá todos los estados de la lavadora, los cuales experimentan ciertas variaciones dependiendo del modo seleccionado. El ciclo transcurre a través de subrutinas de lavado, centrifugado, secado y, finalmente, el estado de finalización, regresando al inicio para permitir la posibilidad de realizar un nuevo ciclo de lavado. Cada una de las subrutinas posee su particular retardo. Durante todo el proceso, se informa continuamente al usuario sobre el estado de la lavadora a través de la consola mediante UART, manteniéndolo al tanto en todo momento.

Para una mejor comprensión del funcionamiento del código, se recomienda la lectura de los comentarios incluidos a lo largo del mismo, disponible en el repositorio de GitHub (Laboratorios/Laboratorio 1/Codigos/CODIGO-LAVADORA-FINAL). Además, se adjunta un vídeo demostrando el funcionamiento óptimo del sistema (Laboratorios/Laboratorio

1/Imágenes, Videos/Link Video en Drive de demostración de Lavadora.txt).

Problema B: Matriz de LEDs

En esta fase de la actividad práctica, se abordó inicialmente la resolución del desplazamiento del mensaje de derecha a izquierda en la matriz de LEDs. Debido al tiempo limitado, se optó por una solución más directa, que consistió en generar cada fotograma del diseño de la matriz y mostrarlo de manera secuencial para crear la visualización del mensaje. Los fotogramas generados se encuentran documentados en el archivo adjunto (Laboratorios/Laboratorio 1/Documentación/FotogramasMensaje.ods), disponible en el repositorio de GitHub relacionado con este informe. Se señala que el mensaje a mostrar es: "Hello World!"

Aislando el funcionamiento del mensaje en movimiento, este opera de la siguiente manera: al iniciar cada fotograma, se carga un registro con un valor definido mediante una macro al inicio del programa. Al finalizar el fotograma, el registro se decrementa y se compara con 0. Si el valor es distinto de cero, el programa vuelve a mostrar el mismo fotograma; una vez que el valor es igual a cero, se procede a mostrar el siguiente fotograma, cargando nuevamente el registro con el valor inicial. El valor definido en la macro permite ajustar la velocidad de desplazamiento, cumpliendo con uno de los requerimientos de la actividad práctica.

Cada fotograma controla las ocho columnas de la matriz, junto con la multiplexación adecuada para seleccionar la fila correspondiente. Este control conjunto permite diseñar el fotograma que se desea mostrar. Además, entre cada diseño de columna, se incluye un retardo que es fundamental para que el fotograma sea visible. Este retardo permite que el encendido de las columnas y filas ocurra a una velocidad tal que, a la percepción humana, se observe una imagen continua.

Una vez implementado correctamente el desplazamiento del mensaje, se investigó una alternativa más eficiente: el uso de una tabla de búsqueda (LUT), donde se almacena el diseño de cada letra. En el programa principal, se seleccionan las letras a mostrar y se desplazan para formar el mensaje. Esta opción tiene ventajas, como la reducción de líneas de código y la posibilidad de modificar fácilmente el contenido del mensaje. Sin embargo, requiere una mayor planificación. Por estas razones, se optó por generar cada fotograma de forma individual, ya que este enfoque permitió completar la actividad en el tiempo estipulado. A pesar de ser más tedioso, el diseño de fotogramas individuales sigue permitiendo futuras modificaciones en el contenido del mensaje, cumpliendo otro de los requisitos de la actividad.

Resuelto el desplazamiento del mensaje, se implementó un menú mediante comunicación UART, que muestra las distintas opciones para la visualización. Para seleccionar una de las seis opciones, se ingresa un número del 0 al 5, y según el número ingresado, el programa accede a una etiqueta correspondiente, mostrando la figura o el mensaje asociado. El funcionamiento para mostrar cada figura (cara feliz, cara triste, corazón, rombo

y alien) es similar al utilizado para visualizar un fotograma individual del mensaje.

Para una mayor comprensión del desarrollo de la actividad, se recomienda revisar el código generado (Laboratorios/Laboratorio 1/Códigos/Codigo-Matriz-FINAL/Codigo-Matriz-FINAL/main.asm) y observar el montaje físico (Laboratorios/Laboratorio 1/Imágenes, Videos/Link Video en Drive de demostración de Lavadora.txt). Ambos archivos están disponibles en el repositorio de GitHub previamente compartido. Cabe destacar que en la implementación física se añadió un pulsador para facilitar la selección de otra figura o mensaje, mejorando así la experiencia del usuario.

VIII. CONCLUSIONES

El laboratorio cumplió con los objetivos establecidos. En primer lugar, se completó con éxito el ensamblaje del modelo de lavadora, integrando todos los componentes necesarios para su control a través del microcontrolador ATmega328P. Además, se desarrolló un sistema de control eficiente para la lavadora automatizada, implementando de manera adecuada los modos de lavado, centrifugado y secado, ajustables según la carga seleccionada, lo que garantiza un funcionamiento adaptable. La comunicación UART permitió el monitoreo en tiempo real y el control remoto del ciclo de lavado, mejorando significativamente la interacción con el sistema.

En cuanto a la matriz de LEDs, se implementó con éxito un mensaje desplazable de forma suave y continua, cumpliendo con los requisitos. A través de la comunicación UART, se habilitó la opción de seleccionar entre diferentes modos de visualización, ya sea el mensaje desplazable o imágenes pre-programadas, lo que añadió flexibilidad y mayor funcionalidad al sistema.

En conclusión, se desarrolló una solución efectiva con Assembler en el ATmega328P, resolviendo los problemas planteados. Tanto el control de la lavadora como el de la matriz de LEDs cumplieron con los objetivos, permitiendo además la posibilidad de realizar ajustes o mejoras en el futuro.

IX. REFERENCIAS BIBLIOGRÁFICAS

- [1] S. F. Barrett, *Embedded system design with the Atmel AVR microcontroller: Part I*, vol. 24. Morgan & Claypool Publishers, 2009. doi: 10.2200/S00225ED1V01Y200910DCS025.
- [2] S. F. Barrett, *Embedded system design with the Atmel AVR microcontroller: Part II*, vol. 25. Morgan & Claypool Publishers, 2009. doi: 10.2200/S00225ED1V01Y200910DCS025.
- [3] S. F. Barrett and D. J. Pack, *Atmel AVR microcontroller primer: Programming and interfacing*, 2nd ed. Morgan & Claypool, 2012.
- [4] S. P. Dandamudi, *Guide to RISC processors for programmers and engineers*, 1st ed. Springer New York, 2005. doi: 10.1007/b139084.
- [5] D. M. Harris and S. L. Harris, *Digital design and computer architecture*. Morgan Kaufmann, 2007.
- [6] W. Kühnel, *AVR RISC microcontroller handbook*, 1st ed. Newnes, 1998.
- [7] A. N. Sloss, D. Symes, and C. Wright, *ARM system developer's guide: Designing and optimizing system software*, 1st ed. Elsevier/Morgan Kaufmann, 2004.