

Laboratorio 1

1st Luis David Nuñez Noguera
Universidad Tecnológica del Uruguay ITRSO
Fray Bentos, Uruguay
luis.nunez@estudiantes.utec.edu.uy

2nd Tiago Corrales Martinez
Universidad Tecnológica del Uruguay ITRSO
Fray Bentos, Uruguay
tiago.corrales@estudiantes.utec.edu.uy

Index Terms—LUT, Plotter, Punzonadora, Asembler

Abstract—Este laboratorio tuvo como objetivo principal la implementación de sistemas de control automatizado mediante el microcontrolador ATmega328P, integrando distintos periféricos para resolver problemas de generación de señales, comunicación y visualización. En la primera parte se desarrolló la lógica de control de una punzonadora con cinta transportadora a través de una máquina de estados finitos, incorporando comunicación serial USART para el manejo remoto de cargas. La segunda parte se centró en la programación de una matriz de LEDs 8x8, en la cual se mostraron mensajes desplazables y figuras predefinidas mediante técnicas de multiplexado. Posteriormente, se implementó un conversor digital-analógico basado en una red resistiva R-2R, acompañado de una tabla de consulta (LUT) para la generación de formas de onda periódicas observadas en el osciloscopio. Finalmente, se programó un plotter controlado por el microcontrolador, capaz de trazar figuras preestablecidas a partir de comandos recibidos por comunicación serial. La práctica permitió aplicar conocimientos de programación en ensamblador, manejo de periféricos y diseño de prototipos funcionales, consolidando competencias propias de la ingeniería mecatrónica.

I. OBJETIVOS

Objetivo General

Desarrollar e implementar sistemas de control automatizado utilizando el microcontrolador ATmega328P, integrando periféricos como sensores, actuadores, matrices de LEDs y convertidores DAC, para la generación de señales, control de procesos y visualización de información en tiempo real, con el fin de fortalecer las competencias en programación en ensamblador, manejo de comunicación serial y diseño de prototipos funcionales en el contexto de la ingeniería mecatrónica.

Objetivos Específicos

- Programar y validar la operación de la punzonadora con cinta transportadora, gestionando distintos modos de funcionamiento y tipos de carga, y estableciendo comunicación USART para monitoreo y control remoto del sistema.
- Implementar un mensaje desplazable y figuras predefinidas en una matriz de LEDs, optimizando la sincronización del desplazamiento y la interacción con el usuario a través de UART, asegurando una visualización clara y ajustable.
- Generar señales analógicas precisas mediante la implementación de una Look-Up Table (LUT) y un DAC R-

2R, visualizando los resultados en un osciloscopio para verificar el comportamiento temporal y la fidelidad de la señal.

II. MATERIALES

A continuación, se describen los materiales empleados en el desarrollo del laboratorio:

- **Kit de montaje Fischertechnik**
 - Estructura modular de cinta transportadora con punzonadora
 - Engranajes, correas y piezas mecánicas
 - Manual de montaje
- **Microcontrolador y electrónica asociada**
 - ATmega328P
 - Pulsadores (inicio y selección de carga)
 - Sensores de posición
 - Motor de la cinta transportadora
 - Motor/solenoide del punzón
 - Resistencias para red DAC R-2R
 - Matriz de LEDs DOLANG
- **Comunicación y señalización**
 - Interfaz USART
 - Computador personal para envío y recepción de comandos
 - LEDs indicativos de estados y cargas
- **Instrumentos de medición y apoyo**
 - Osciloscopio digital
 - Fuente de alimentación de laboratorio
 - Cables de conexión

III. MARCO TEÓRICO

Antes de detallar el procedimiento llevado a cabo, es necesario clarificar distintos conceptos importantes para la comprensión total de la práctica.

A. ATmega328p

Un microcontrolador puede definirse como un circuito integrado que incorpora en un solo dispositivo una unidad central de procesamiento (CPU), memoria volátil y no volátil, así como periféricos de entrada y salida programables. A diferencia de los microprocesadores de propósito general, los microcontroladores están diseñados para ejecutar tareas específicas de control en sistemas electrónicos.

Este tipo de dispositivos constituye el núcleo de los sistemas embebidos, los cuales se caracterizan por estar dedicados a una función concreta dentro de un sistema mayor. Dichos sistemas se encuentran presentes en una amplia variedad de aplicaciones cotidianas y de carácter industrial, tales como electrodomésticos, automóviles, dispositivos médicos, equipos de telecomunicaciones y sistemas de automatización. Su relevancia radica en que permiten el diseño de soluciones con bajo consumo energético, tamaño reducido, costo accesible y alta confiabilidad.

1) *Características básicas del ATmega328P*: El ATmega328P, fabricado por Microchip Technology (anteriormente Atmel), es uno de los microcontroladores más representativos de la arquitectura AVR de 8 bits. Es ampliamente conocido por ser el componente central de placas de desarrollo como la Arduino Uno, lo que lo ha convertido en un referente tanto en entornos educativos como en aplicaciones prototípicas y comerciales.

Entre sus principales características técnicas destacan:

- Arquitectura AVR RISC de 8 bits, que permite ejecutar la mayoría de instrucciones en un solo ciclo de reloj y optimizar el rendimiento mediante un pipeline sencillo.
- Memoria integrada, compuesta por 32 KB de memoria Flash programable, 2 KB de SRAM y 1 KB de EEPROM, lo que posibilita almacenar tanto el programa como datos temporales y permanentes.
- Puertos de entrada y salida digital (GPIO), con un total de 23 líneas configurables, capaces de manejar señales de entrada o salida y de activar resistencias pull-up internas.
- Módulos temporizadores y contadores, que incluyen dos de 8 bits y uno de 16 bits, con soporte para funciones de modulación por ancho de pulso (PWM).
- Interfaces de comunicación serial, como USART (síncrona y asíncrona), SPI (Serial Peripheral Interface) y TWI (I²C), lo que facilita la conexión con otros microcontroladores o periféricos externos.
- Conversor analógico-digital (ADC) de 10 bits con hasta seis canales, que posibilita la digitalización de señales analógicas.
- Capacidad de operación hasta los 20 MHz y disponibilidad de múltiples modos de bajo consumo, lo cual lo hace adecuado para aplicaciones portátiles y autónomas.

2) *Ventajas del uso de lenguaje ensamblador frente a lenguajes de alto nivel en el control de hardware*: La programación de microcontroladores puede realizarse mediante lenguajes de alto nivel, como C o C++, o directamente en lenguaje ensamblador. Si bien los lenguajes de alto nivel ofrecen mayor facilidad de programación, portabilidad y legibilidad del código, el ensamblador conserva ventajas significativas en aplicaciones donde se requiere un control exhaustivo del hardware.

Entre sus principales beneficios destacan los siguientes:

- Permite un acceso directo a los registros internos y periféricos del microcontrolador, sin capas de abstracción.

- Favorece la optimización del tiempo de ejecución, dado que el programador puede seleccionar instrucciones específicas y aprovechar al máximo cada ciclo de reloj.
- Genera programas más compactos, lo que resulta crucial en dispositivos con recursos de memoria limitados.
- Proporciona mayor precisión temporal, aspecto fundamental en aplicaciones de tiempo real donde el cumplimiento de plazos estrictos es determinante.

No obstante, el uso exclusivo de ensamblador puede resultar poco práctico en proyectos de gran envergadura, debido a su complejidad, menor portabilidad y mayor dificultad en el mantenimiento del código. Por ello, en la práctica actual es común emplear un enfoque mixto: desarrollar la mayor parte del programa en un lenguaje de alto nivel y reservar el ensamblador para las rutinas críticas que demandan máxima eficiencia.

B. Comunicación Serial (USART)

La comunicación serial es un método de transmisión de datos en el cual la información se envía bit a bit a través de un canal de comunicación. Dentro de esta categoría, la interfaz UART (Universal Asynchronous Receiver and Transmitter) y su extensión USART (Universal Synchronous and Asynchronous Receiver and Transmitter) son ampliamente utilizadas en microcontroladores y sistemas embebidos.

En el modo asíncrono, la transmisión no requiere una señal de reloj compartida entre emisor y receptor. En su lugar, ambos dispositivos deben acordar previamente parámetros como la velocidad de transmisión (baud rate) y el formato de la trama de datos. La sincronización se logra mediante bits especiales que indican el inicio y fin de cada paquete transmitido.

Por otro lado, en el modo síncrono (propio de USART), se utiliza una línea de reloj adicional, lo cual permite mayor velocidad y confiabilidad en la comunicación. Sin embargo, el modo asíncrono UART sigue siendo el más utilizado debido a su simplicidad y menor requerimiento de pines.

1) *Formato de transmisión*: La transmisión de datos mediante UART/USART sigue un formato estructurado de trama, compuesto por los siguientes elementos:

- **Bit de inicio (Start bit)**: Indica el comienzo de la transmisión; normalmente consiste en un nivel lógico bajo (0).
- **Bits de datos**: Generalmente entre 5 y 9 bits, aunque lo más común es el uso de 8 bits, lo que permite la transmisión de un byte por trama.
- **Bit de paridad (opcional)**: Permite la detección básica de errores en la transmisión, pudiendo configurarse como par, impar o sin paridad.
- **Bit(s) de parada (Stop bit)**: Señalizan el final de la trama; pueden configurarse en 1 o 2 bits.
- **Velocidad de transmisión (baud rate)**: Corresponde al número de símbolos transmitidos por segundo. En UART, un baud equivale a un bit, por lo que baud rate y bit rate son equivalentes. Valores comunes son 9600, 19200 y 115200 baudios, dependiendo de la aplicación.

La correcta configuración de estos parámetros en ambos extremos es fundamental para garantizar una comunicación confiable y libre de errores.

C. Máquinas de Estados

Una máquina de estados finitos (FSM, por sus siglas en inglés Finite State Machine) es un modelo matemático y computacional que describe el comportamiento de un sistema en términos de estados y transiciones. Un estado representa una condición particular en la que se encuentra el sistema, mientras que las transiciones definen los cambios de un estado a otro en respuesta a determinados eventos o condiciones de entrada.

Este enfoque resulta ampliamente utilizado en la automatización y control de procesos, ya que permite representar de manera clara y sistemática el flujo lógico de operaciones que debe ejecutar un sistema embebido o un controlador industrial. En términos prácticos, una FSM se compone de:

- Un conjunto finito de estados posibles.
- Un estado inicial desde el cual comienza la ejecución.
- Reglas de transición que definen el paso entre estados.
- Posibles estados finales que indican la culminación del proceso.

D. Conversores Digital–Analógico (DAC R-2R)

1) *Principio de funcionamiento del DAC resistivo R-2R:* Un conversor digital–analógico (DAC, por sus siglas en inglés Digital to Analog Converter) es un dispositivo cuya función es transformar una señal digital, representada mediante combinaciones binarias, en una señal analógica continua en el tiempo. Entre los distintos tipos de DAC, el de red resistiva R-2R se distingue por su simplicidad, precisión y facilidad de implementación en circuitos electrónicos.

El DAC R-2R está constituido por una red de resistencias organizadas en una topología de escalera, donde solo se emplean dos valores: una resistencia de valor R y otra de valor 2R. La configuración de esta red permite dividir la tensión de referencia en fracciones binarias, de manera que cada bit de entrada digital controla un interruptor electrónico que conecta un nodo de la red a un nivel lógico alto o bajo. De esta forma, la contribución de cada bit se pondera de acuerdo con su peso binario, garantizando una conversión lineal entre la palabra digital y la señal analógica obtenida.

2) *Relación entre señal digital y voltaje analógico de salida:* La salida de un DAC R-2R puede expresarse como una fracción proporcional de la tensión de referencia V_{ref} . Para un conversor de N bits, la relación entre la palabra digital de entrada y el voltaje de salida V_{out} está dada por:

$$V_{out} = V_{ref} \cdot \frac{D}{2^N} \quad (1)$$

Donde D representa el valor decimal de la palabra binaria aplicada al DAC y 2^N es el número total de niveles posibles. De esta manera, la resolución del conversor está determinada por el número de bits, lo cual define el tamaño del incremento mínimo de tensión o Least Significant Bit (LSB).

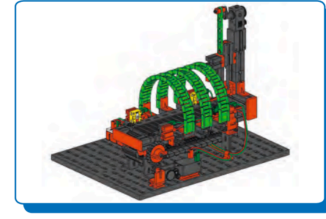


Fig. 1. Referencia del modelo de cinta transportadora con punzadora

En consecuencia, a mayor número de bits, mayor será la resolución y menor la diferencia entre niveles consecutivos de salida, lo que se traduce en una señal analógica más precisa y cercana al valor ideal.

3) *Uso en generación de señales periódicas mediante LUT:* Una aplicación común de los DAC R-2R en sistemas embebidos es la generación de señales periódicas, como ondas senoidales, triangulares o cuadradas. Para ello, se emplea el método de la tabla de consulta (Look-Up Table, LUT).

En este enfoque, los valores discretizados de una señal periódica se almacenan previamente en memoria en forma de una tabla. El microcontrolador lee secuencialmente estos valores digitales y los envía al DAC a una frecuencia determinada. Como resultado, el conversor produce en su salida una señal analógica que reconstruye la forma de onda deseada.

Este procedimiento permite implementar generadores de funciones, síntesis de audio, modulación y control de actuadores, con una complejidad relativamente baja en hardware y software. La calidad de la señal generada dependerá de la resolución del DAC, de la precisión de la LUT y de la frecuencia de actualización empleada.

IV. PROCEDIMIENTO

Para llevar a cabo la práctica, se siguieron los siguientes pasos generales y específicos por cada circuito:

A. Parte A

En primer lugar, se procedió a la lectura de la documentación básica del kit Fischertechnik, con el objetivo de familiarizarse con los componentes disponibles y las recomendaciones de montaje. Posteriormente, se revisaron las instrucciones del manual de construcción, donde se obtuvieron las pautas necesarias para el armado y cableado del modelo.

Una vez comprendidas las indicaciones, se realizó el montaje de la cinta transportadora con punzonadora, siguiendo paso a paso las guías del kit. Una imagen de referencia puede verse en la figura 1.

Durante el armado, se tuvo especial cuidado en la conexión de los motores y sensores al microcontrolador ATmega328P, prescindiendo del controlador incluido en el kit y empleando únicamente el hardware definido en la consigna.

Con el modelo ensamblado, se procedió a la programación en lenguaje ensamblador del ATmega328P. En esta etapa, se desarrolló la lógica de control a partir de una máquina de estados finitos, la cual contempló los modos de funcionamiento requeridos: alimentación, punzonado y descarga. Cada transición

de estado fue gestionada mediante temporizadores internos del microcontrolador, asegurando los tiempos establecidos para cada tipo de carga (ligera, media y pesada).

A continuación, se implementó la comunicación serial USART, permitiendo la recepción de comandos. Se verificó el funcionamiento remoto mediante la transmisión de la orden de inicio y la selección de cargas a través de caracteres enviados desde un sistema externo.

B. Parte B

La segunda parte del laboratorio consistía en la utilización de una matriz de LEDs 8x8 para mostrar una frase y unas figuras trabajadas en tareas de evaluación continua anteriormente. La implementación en físico de este apartado no radicaba de mucha complejidad, con la matriz a disposición se debieron ubicar los pines que manejan las filas y las columnas (2). Esto es posible con el datasheet del componente o simplemente probar pin a pin para ver que hacía. Separados los pines entre filas y columnas se procedió a la conexión en la placa del microcontrolador. Se usaron los puertos B, C y D con los pines PC2, PC3, PD2, PD3, PD4, PD5, PD6, PD7 para el manejo de las columnas o LEDs a prender y los pines PB0, PB1, PB2, PB3, PB4, PB5, PC0, PC1 para el manejo de las filas. Con esos aspectos básicos definidos se pasó a la creación del código que se mencionará más adelante en el documento. Este último sería el encargado de gestionar el encendido multiplexado de la matriz, la selección dinámica de filas y columnas, y la sincronización necesaria para que las letras de la frase y las figuras se visualizaran de manera correcta y continua.

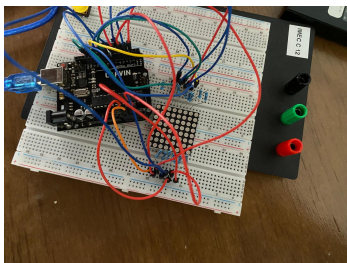


Fig. 2. Matriz LEDs implementación física

C. Parte C

La tercera parte del laboratorio consistía en un conversor digital a analógico 2R-R (3). Donde este radica en el uso solamente de resistencias para hacerlo funcionar. Para este tipo de configuración el número de resistencias 'R' tiene que ser el número de bits de cada dato. Se trabajaron con datos de ocho bits, entonces se colocaron ocho resistencias con el valor 'R'. El valor elegido de las resistencias fue de 1 K Ω , por consiguiente, el valor para las resistencias que conforman al grupo '2R' son de 2 K Ω . La cantidad de estas resistencias '2R' usadas fueron de nueve.

Para el armado y disposición del circuito nos guiamos en la información brindada por los profesores. Donde las resistencias denominadas 'R' se conectaron en serie entre sí,

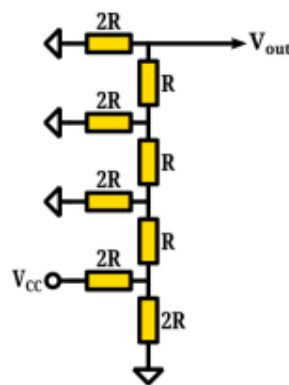


Fig. 3. Conversor DAC 2R-R

aunque la primera resistencia en esa cadena es de valor '2R', y en cada nodo de estas sale una resistencia '2R' conectada a tierra o al pin de la placa para manejar el funcionamiento. El osciloscopio fue conectada a la última resistencia de valor 'R' y a tierra para ver la onda descrita por la señal brindada en la letra del laboratorio. Como se mencionó anteriormente a las salidas de las resistencias '2R' conectadas en los nodos de la cadena de resistencias 'R' se le conectan los pines de la placa para el funcionamiento del modelo. Donde el bit más significativo se conecta en la resistencia de valor '2R' más cercana a la resistencia donde se conecta el osciloscopio y por consiguiente el bit menos significativo se conecta a la resistencia de valor '2R' más cercana al comienzo de la cadena de resistencias de valor 'R'.

Con las conexiones correctamente hechas y el código bien implementado se debería ver la señal esperada (4), aunque pueda tener ruido porque no son ideales los materiales usados.

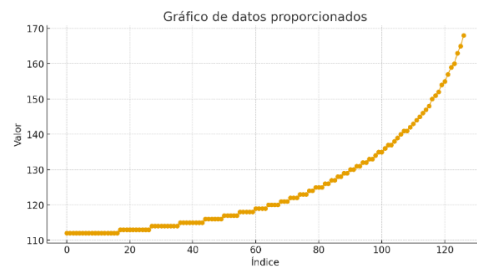


Fig. 4. Gráfico señal

D. Parte D-E

Finalmente, se programó el plotter controlado por el microcontrolador. Se estableció la comunicación con el sistema a través de comandos seriales, lo que permitió seleccionar y ejecutar el trazado de figuras predefinidas (triángulo, cruz y conjunto de todas). Para el control de los actuadores, se utilizaron relés/MOSFET como interfaz entre el microcontrolador y el PLC del plotter.

Una imagen del plotter a utilizar puede verse en la figura 5

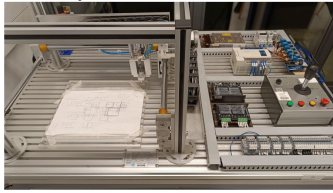


Fig. 5. Plotter del Laboratorio de Mecatrónica

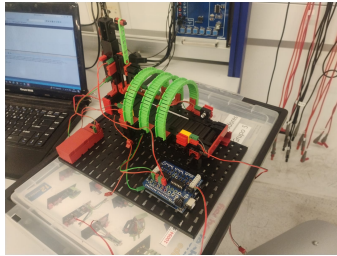


Fig. 6. Modelo de Cinta Transportador con Punzonadora

V. RESULTADOS

A. Parte A

En primer lugar, el modelo físico de la cinta transportadora con punzonadora fue armado con éxito, este puede divisarse en la figura 6. En las pruebas iniciales, se observó que el sistema respondió adecuadamente a los diferentes modos de carga (ligera, media y pesada), cumpliendo con los tiempos programados de avance, pausa y punzonado. La máquina de estados implementada en el código en ensamblador permitió transiciones fluidas entre los estados de espera, alimentación, punzonado y descarga.

En la fase de inicialización, se configuran la pila, la comunicación serial USART (a 9600 baudios) y los puertos de entrada/salida necesarios para controlar motores. Posteriormente, el programa envía un mensaje inicial por el puerto serial indicando las opciones de carga disponibles: ligera, mediana y pesada.

El sistema espera la recepción de un comando ("L", "M" o "P"), y en función de ello ejecuta la rutina correspondiente. Cada rutina (ligera, mediana o pesada) controla el avance de la cinta transportadora y la activación del punzón con los tiempos establecidos en la consigna:

- Ligera: avances y pausas cortos.
- Mediana: tiempos intermedios.
- Pesada: tiempos más largos de avance y presión del punzón.

Para ello, el programa utiliza subrutinas de temporización (s1Delay y ms200Delay) implementadas mediante bucles de decremento, que generan retardos aproximados en segundos y milisegundos. Una vez completado el ciclo, el sistema vuelve a enviar el mensaje inicial, quedando listo para recibir un nuevo comando.

B. Parte B

El programa desarrollado tiene como objetivo mostrar mensajes y figuras en una matriz de LEDs utilizando un microcontrolador AVR. Para ello, se configura la UART, que permite la comunicación serial, y se inicializan los puertos necesarios para controlar las filas y columnas de la matriz. La idea principal del código es permitir que el usuario, mediante datos enviados por UART, pueda elegir si desea visualizar un mensaje o una figura predefinida.

El mensaje principal, codificado en la memoria de programa, corresponde a la frase "JESUS MOLESTO" y se despliega columna por columna en la matriz mediante un sistema de multiplexado, que permite que los LEDs se enciendan de manera rápida y continua, generando la impresión de una imagen estable. De forma similar, el usuario puede seleccionar entre varias figuras, como una cara feliz, una cara triste, un corazón, un rombo o un alien, las cuales también se muestran usando el mismo mecanismo de multiplexado.

El código incluye subrutinas específicas para transmitir datos por UART, actualizar la posición del mensaje, controlar cada columna de la matriz y apagar los LEDs cuando es necesario. En conjunto, estas funciones permiten que el microcontrolador gestione tanto la entrada del usuario como la salida visual en la matriz, logrando que el mensaje o la figura seleccionada se muestre de manera clara y continua.

Lamentablemente el código proporcionado no tiene funcionalidad, no se pudo hacer que este funcionara para cumplir con la consigna, asimismo los conocimientos adquiridos son de utilidad para las próximas entregas.

C. Parte C

Pasando al código de esta tercera parte del laboratorio, primero se configuran todos los pines del puerto D como salidas y se inicializa el stack pointer. Se utiliza el Timer2 con un prescaler específico para generar interrupciones periódicas que controlan la frecuencia de actualización de la señal.

Cada interrupción ejecuta la rutina RSI, que toma el siguiente valor de la tabla LUT y lo envía al puerto D. La tabla define la forma de la señal que se genera, y un contador recorre secuencialmente estos valores. Cuando se alcanza el final de la tabla (Final), el contador se reinicia, haciendo que la señal se repita de manera continua.

El bucle principal del programa permanece vacío, ya que toda la generación de la señal depende de las interrupciones, lo que permite un control preciso de la temporización y de la forma de la onda sin bloquear el microcontrolador. De este modo, el usuario puede observar la señal en un osciloscopio y verificar la secuencia de cambios definidos en la LUT, lo que facilita el análisis de la frecuencia y forma de onda (7). Se obtuvo la señal esperada de manera satisfactoria correspondiendo al gráfico anteriormente realizado (4).

D. Parte D-E

Por último, en el plotter controlado por el microcontrolador, se comprobó el correcto funcionamiento del sistema de trazado. Las figuras básicas (triángulo y cruz) se dibujaron

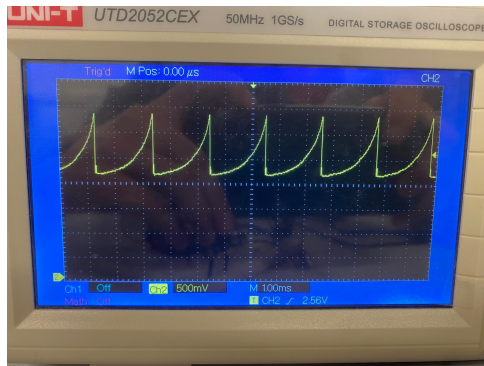


Fig. 7. Señal resultante conversor DAC 2R-R

dentro del área delimitada, respetando los tiempos de conmutación de los relés. El comando “T” permitió ejecutar de forma secuencial el trazado de todas las figuras.

Con respecto al código, en la inicialización, se configura la pila, el puerto USART y los pines de salida conectados a los actuadores del plotter (a través de relés/MOSFET). Una vez listo, el sistema envía un menú de opciones al monitor serial, mostrando las figuras disponibles:

- Triángulo
- Cruz
- T. Todas las figuras en secuencia

Los dibujos realizados pueden verse en las figuras 8 y 9 para el triángulo y la cruz respectivamente.

El código queda en espera de un carácter ingresado por el usuario y, según la selección, llama a la rutina correspondiente. Cada rutina activa las salidas digitales necesarias para generar los movimientos del plotter en los ejes X e Y, así como el control del solenoide encargado del trazado.

El programa incluye subrutinas auxiliares como centrar y descentrar, que permiten posicionar el lápiz en la hoja antes y después del dibujo, asegurando precisión y orden en la ejecución.

De esta manera, el código permitió la automatización del proceso de dibujo, recibiendo órdenes desde el puerto serial y ejecutando trazos predefinidos. El diseño modular, basado en rutinas, facilita la incorporación de nuevas figuras en el futuro.

VI. CONCLUSIÓN

La realización de este laboratorio permitió aplicar de forma práctica los conceptos teóricos de programación en ensamblador y control de periféricos mediante el microcontrolador ATmega328P. A lo largo de las distintas etapas, se logró concretar la programación y puesta en marcha de la punzonadora con cinta transportadora, gestionando de manera correcta los distintos modos de carga y tiempos de operación. Asimismo, se consiguió establecer comunicación serial USART, posibilitando el control remoto del sistema, lo que consolidó el entendimiento de la interacción entre hardware y software en sistemas embebidos.

En relación con la matriz de LEDs, si bien se alcanzó a comprender y estructurar el código para el desplazamiento de

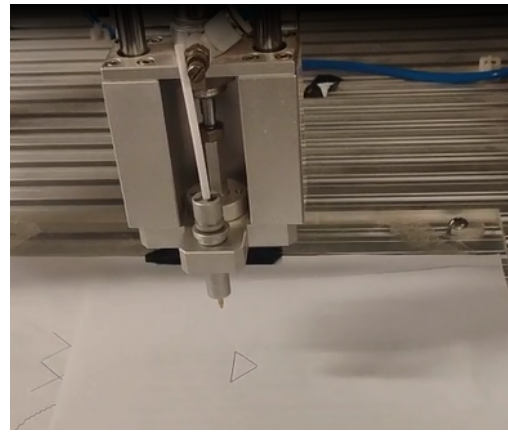


Fig. 8. Plotter - Figura: Triángulo



Fig. 9. Plotter - Figura: Cruz

mensajes y la visualización de figuras, el mismo no logró la funcionalidad esperada en la práctica. A pesar de esta limitación, la experiencia adquirida resulta valiosa para futuras implementaciones, al permitir identificar errores y comprender con mayor profundidad las técnicas de multiplexado y sincronización.

En la tercera parte, la implementación del DAC R-2R junto con una tabla de consulta (LUT) fue exitosa, obteniéndose en el osciloscopio una señal periódica que correspondió a la forma de onda planificada. Esto permitió verificar en la práctica la relación entre datos digitales y señales analógicas, reafirmando la utilidad de esta técnica en aplicaciones de generación de señales.

Finalmente, en la etapa correspondiente al plotter, se comprobó el correcto trazado de figuras del triángulo y de la cruz, más no el círculo. Dichas figuras eran seleccionadas a través

de comunicación serial. La estructura modular del código y las rutinas auxiliares implementadas facilitaron la automatización del dibujo, demostrando la aplicabilidad del microcontrolador en sistemas de control de mayor complejidad.

VII. BIBLIOGRAFÍA

REFERENCES

- [1] R. J. Tocci, N. S. Widmer, G. L. Moss, *Sistemas digitales: Principios y aplicaciones*, 10ma Edición, traducción y revisión técnica de R. F. Acuña y M. de Alba, Pearson Educación de México, S.A. de C.V., 2007. ISBN: 9702609704, ISBN-13: 9789702609704.