Diseño e Implementación de un Reloj Digital sobre Arduino MEGA con Programación en Lenguaje Escalera

Jiménez García José de Jesús
Escuela de Ingeniería
Universidad La Salle Pachuca
Pachuca , Hidalgo
jose.jimenez@lasallep.mx

Ordaz Oliver Mario Oscar
Departamento de Ingeniería Eléctrica y Electrónica
Tecnológico Nacional De México, Campus Pachuca,
Pachuca Hidalgo
mario.oo@pachuca.tecnm.mx

Gutiérrez Moreno Evelin
Ingeniería Mecatrónica
Universidad Politécnica de Pachuca,
Pachuca Hidalgo
evgutierrez@upp.edu.mx

Montiel Hernández Justo Fabián
Escuela Superior Ciudad Sahagún
Universidad Autónoma del Estado de Hidalgo
Cd. Sahagún, México
justo_montiel@uaeh.edu.mx

Espejel Rivera María Angélica
Departamento de Ingeniería Eléctrica y Electrónica
Tecnológico Nacional De México, Campus Pachuca,
Pachuca, Hidalgo
electricayelectronica@pachuca.tecnm.mx

Ordaz Oliver Jesús Patricio
Centro de Investigación en tecnologías de la información y Sistemas,
Universidad Autónoma del Estado de Hidalgo,
Pachuca Hidalgo
jesus_ordaz@uaeh.edu.mx

Resumen: **En este trabajo se presenta el diseño y la programación de un reloj digital utilizando lenguaje escalera, implementado en un microcontrolador Arduino Mega y visualizado en cuatro displays de 7 segmentos. El objetivo principal del proyecto es desarrollar un reloj que muestre horas y minutos en formato de 12 horas, empleando exclusivamente bobinas temporizadas con contactos de retardo a la conexión. A diferencia de los enfoques convencionales en electrónica digital, donde se utilizan flip-flops para la implementación de circuitos secuenciales, esta propuesta explora la viabilidad y eficiencia de emplear diagramas en lenguaje escalera junto con bobinas temporizadas. El trabajo detalla los procedimientos de diseño del diagrama en lenguaje escalera, la integración y conexión de los distintos componentes de hardware, la programación del microcontrolador Arduino, y los resultados obtenidos a partir de las pruebas realizadas. Además, se discuten en profundidad las ventajas y limitaciones identificadas en el proceso, así como las posibles mejoras para optimizar el rendimiento y la precisión del reloj.**

Palabras clave—, ***ATmega2560, cascada, contador, Display, flip-flop, Ladder Maker, prototipo, salidas, temporizador.***

I. Introducción

Los relojes digitales son dispositivos presentes en una gran cantidad eventos de nuestra vida cotidiana, encontrándose en aplicaciones que van desde dispositivos personales hasta sistemas industriales. El diseño y programación de relojes digitales tradicionalmente involucra el uso de flip-flops para la implementación de circuitos secuenciales [1]. Sin embargo, esta metodología puede resultar compleja y menos intuitiva para quienes no están familiarizados con la electrónica digital. La motivación de este proyecto radica en explorar las capacidades del lenguaje escalera y su integración con hardware de Arduino, proporcionando una solución precisa y eficiente para la visualización del tiempo en un formato de 12 horas. La elección del Arduino Mega se justifica por su amplio número de salidas, lo que lo hace ideal para manejar múltiples Displays de 7 segmentos. Utilizar diagramas escalera y bobinas temporizadas presenta una alternativa viable y efectiva a los flip-flops, simplificando el diseño y la implementación de circuitos secuenciales.

El uso del lenguaje escalera es común en la automatización industrial debido a su facilidad de lectura y capacidad para representar lógicas de control complejas de manera gráfica. Estudios previos han demostrado la efectividad de este lenguaje en aplicaciones de control de procesos y monitoreo industrial. Además, la integración de hardware como Arduino en proyectos de control ha ganado popularidad gracias a su versatilidad y bajo costo. La implementación de circuitos secuenciales con diagramas escalera y bobinas temporizadas ha sido menos explorada en comparación con los métodos tradicionales que utilizan flip-flops, lo que resalta la innovación y relevancia de este proyecto [2].

Este proyecto presenta una serie de contribuciones prácticas asociadas con los circuitos secuenciales y la electrónica digital, entre las que se destacan las siguientes:

i) Implementar de forma práctica el lenguaje escalera para programar un reloj digital en un entorno grafico de Arduino.
ii) Diseñar una metodología clara para la configuración de temporizadores y asignación de salidas en un Arduino Mega.
iii) Discutir los resultados de pruebas y validaciones que confirman la viabilidad y precisión del reloj digital programado.
iv) Interpretar los resultados y validar la viabilidad de la propuesta para sustituir a los flip-flops, mediante el

uso de diagramas escalera y bobinas con contactos temporizados.

v) Crear las bases para futuras investigaciones y mejoras en la integración de sistemas de control industriales con hardware de desarrollo.

De acuerdo con los objetivos, alcances y limitaciones, el presente documento se organiza en cinco secciones principales. Inicialmente en la sección I, se presenta la contextualización del problema, misma que permite justifica la investigación presente, se presenta una revisión la literatura relevante, se describen las contribuciones del proyecto y se presenta la organización general de las secciones del documento. Posteriormente en la sección II**,** se describe a detalle el hardware y software utilizados, así como el proceso de programación del reloj digital. En la sección III, se presentan los hallazgos y resultados obtenidos durante el desarrollo y pruebas del reloj digital. Corresponde a la sección IV, la discusión, análisis de las ventajas, limitaciones y posibles mejoras del proyecto. Finalmente, en la sección V, se resumen los hallazgos principales y la relevancia del proyecto.

II. Metodología

En esta sección se realiza una descripción detallada de los componentes principales del reloj digital construido para el proyecto; así como las conexiones entre las salidas de la tarjeta de desarrollo programable Arduino Mega y los Displays de 7 segmentos, por medio de los codificadores BCD 47LS74.

Arduino es una plataforma de hardware libre basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares [3]. Es ampliamente utilizada en robótica y proyectos de automatización debido a su simplicidad y flexibilidad para conectar y programar dispositivos, tales como sensores, motores y luces. El Arduino Mega es una placa de desarrollo basada en el microcontrolador ATmega2560, que ofrece más pines de entrada/salida y mayor memoria que otros modelos, misma que tiene las siguientes características principales [4]:

- Microcontrolador: ATmega2560
- Voltaje de operación: 5v
- Pines digitales de IE/S: 54 (15 PWM)
- Pines analógicos de entrada: 16
- Memoria Flash: 256 KB
- SRAM: 8 KB
- EEPROM: 4 KB

Un Display de 7 segmentos es un dispositivo de visualización electrónica que se usa para mostrar dígitos decimales. Cada dígito se compone de 7 LEDs individuales (segmentos), los cuales pueden ser encendidos o apagados para formar números o letras. El encapsulado 74LS47 es un circuito integrado decodificador de 7 segmentos. Un decodificador es un dispositivo que convierte una secuencia de entradas binarias en una salida más grande con un número de salidas más pequeño. El 74LS47 es un decodificador de 7 segmentos, lo que significa que tiene 8 entradas y 7 salidas, el cual permite codificar una entrada binaria de 4 bits a una salida BCD visible en un Display de 7 segmentos, es decir, el decodificador BCD a 7 segmentos convierte una entrada en código binario decimal (BCD) en una señal adecuada para controlar un Display de 7 segmentos, permitiendo la visualización de números del 0 al 9 en formato decimal.

El lenguaje en escalera, o diagrama de escalera, es una representación gráfica de los circuitos eléctricos que se asemeja a los esquemas de las escaleras, donde los dispositivos de control y de maniobra se conectan por líneas que representan los conductores eléctricos. Este lenguaje es común en la programación del PLC (controlador lógico programable) y simula la lógica utilizada en relés eléctricos, siendo muy intuitivo para los ingenieros eléctricos y fácilmente adaptable a proyectos de automatización.

Para la construcción de este primer prototipo se plantea el uso de una placa de pruebas (protoboard), ya que permite tanto construir, como examinar circuitos electrónicos y realizar conexiones eléctricas sin necesidad de soldadura. Sabiendo que la resistencia es una medida de la oposición al flujo de corriente en un circuito eléctrico, es necesario utilizar un conjunto de resistencias eléctricas de 0.25 watts y 220Ω para limitar la corriente que fluye a través de cada segmento en los Displays.

A. Materiales

En la Tabla I. se muestran los principales componentes utilizados para el proyecto, dicha tabla incluye la cantidad y los costos (en pesos mexicanos) aproximados de cada componente.

TABLA I
LISTA DE PRECIOS

Cantidad	Componente	Precio por unidad
4	Display de 7 segmentos	$45
28	Resistencias	$15
2	Protoboard	$200
1	Arduino mega	$350
1	Jumpers	$50 (paquete)
4	CI 74LS47	$60
1	MDF 5.5 mm	$126 m^2
1	Impresión 3D	$100

B. Pasos para la construcción

Ladder Maker es un software especializado de acceso libre para el diseño y programación en lenguaje escalera, permitiendo la visualización gráfica de diagramas y la carga directa en varios modelos de Arduino. Facilita la creación y simulación de circuitos de control para aplicaciones de automatización, especialmente en entornos industriales. Este software permite programar controladores lógicos programables (PLC), como el microcontrolador ATmega2560, mediante un entorno visual intuitivo que representa contactos y bobinas en un esquema de diagrama escalera, facilitando el desarrollo de sistemas de control secuenciales y combinacionales comunes en manufactura y automatización [5].

Como primer tarea en la parte del proceso de programación, se diseñó un diagrama que representa la lógica de control para

un reloj de 12 horas, mediante el uso de bobinas temporizadas con contactos con retardo a la conexión para controlar la secuencia de los segmentos de los Displays. Posteriormente y de acuerdo con el mapa de E/S de Ladder Maker, se asignaron los pines del Arduino Mega a cada entrara de los decodificadores BCD 74LS47, esto para registrar cada uno de los segmento de los Displays de manera que se pueda controlar individualmente cada uno de ellos. Después se revisó detalladamente la configuración de los temporizadores en el diagrama de escalera para asegurar que los Displays cambien de estado en los intervalos de tiempo correctos. La utilización de temporizadores para generar los pulsos necesarios que incrementen las horas y minutos. El siguiente paso en el proceso consiste en el uso de Ladder Maker para cargar el diagrama escalera directamente en la memoria del Arduino Mega y comprobar mediante pruebas preliminares que su funcionamiento sea correcto. Dichas pruebas dieron lugar a una serie de ajustes en los temporizadores y en la lógica del diagrama, lo cual permitió una validación final del funcionamiento de la programación y de las conexiones, asegurando precisión y estabilidad en el reloj.

III. Resultados y Discusión

En esta sección se presentan los detalles principales asociados con la construcción y programación del reloj digital propuesto para este proyecto, de manera similar, se plantean algunos de los retos y limitaciones a los que se enfrentaron los autores en el desarrollo de este. Uno de los puntos más importantes por mencionar es, que el tipo de Display utilizado en la construcción del reloj propuesto es de ánodo común, el cual es compatible con el decodificador digital 47LS74. Dada esta elección, se recomienda revisar la hoja de datos tanto del Display como del decodificador BDC para conocer la ubicación correcta de los segmentos del decodificador con respecto a los pines de entrada del Display [6].

Para analizar la lógica del reloj digital que se va a utilizar al momento de programar en el software Ladder Maker, se considera la aplicación de banderas, conocidas también como bobinas internas, con contactos asociados a dichas bobinas y temporizadores en cascada, dichos elementos permiten llevar una secuencia tal como se requiere en la lógica de funcionamiento del reloj con formato de 12 horas.

El uso principal de banderas (bobinas internas) es manejar los estados intermedios y la activación secuencial de los temporizadores, para que los contactos asociados permitan la validación de los estados de las bobinas y los temporizadores. Cabe mencionar que por medio del decodificador 74LS47 es posible controlar un Display de 7 segmentos a través de 4 salidas del Arduino Mega, simplificando la lógica que se programa en Ladder Maker.

Durante las pruebas, surgieron algunos desafíos significativos, puesto que inicialmente el programa fue diseñado con el uso de contadores, lo que permitía una simplificación considerable del programa y de la lógica de este, sin embargo, esta metodología no permitió el reinicio de la secuencia al finalizar el primer ciclo. La primer limitante y restricción detectada en el uso de temporizadores con Ladder Maker es que el tiempo máximo que puede cronometrar un temporizador es de 10 minutos, lo cual modifica significativamente la lógica de las horas respecto a la de los minutos, es decir, se utilizan 6 temporizadores en cascada (cada uno de 10 minutos) para construir una hora, como se muestra en la Figura (1). La segunda restricción identificada en Ladder Maker es que las salidas 17, 18, 19, 20, 21 y 22 no están físicamente acopladas a la programación en escalera de Ladder Maker como lo indica su diagrama E/S, situación que restringe el reloj a usar únicamente 4 Displays, sin la posibilidad de agregar 2 leds con intermitencia para indicar el paso de los segundos.

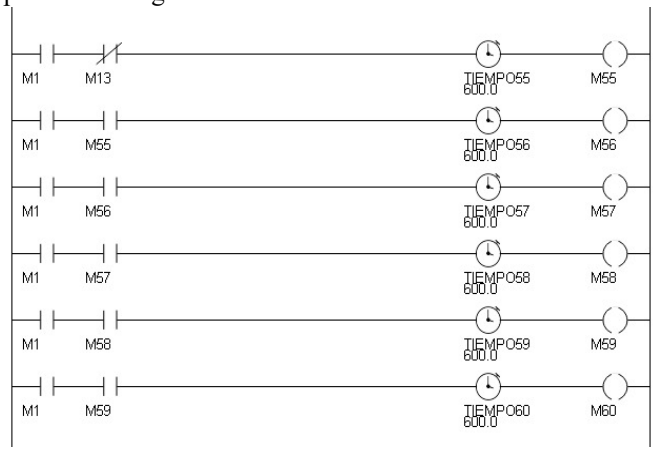

Fig. 1 Diagrama escalera de los temporizadores en cascada para el conteo de horas.

Cabe destacar que, en la configuración en cascada de los temporizadores, el primero activa al segundo, el segundo al tercero y así sucesivamente, y para cuando el último temporizador finaliza, este reinicia al primero, dicha acción desata un efecto en cadena que es utilizado para reiniciar el ciclo e incrementar la temporización de las horas. Aunque la implementación de temporizadores en cascada puede resultar un compleja, es una herramienta que simplifica la lógica del reloj digital al ser utilizado el lenguaje de diagramas en escalera.

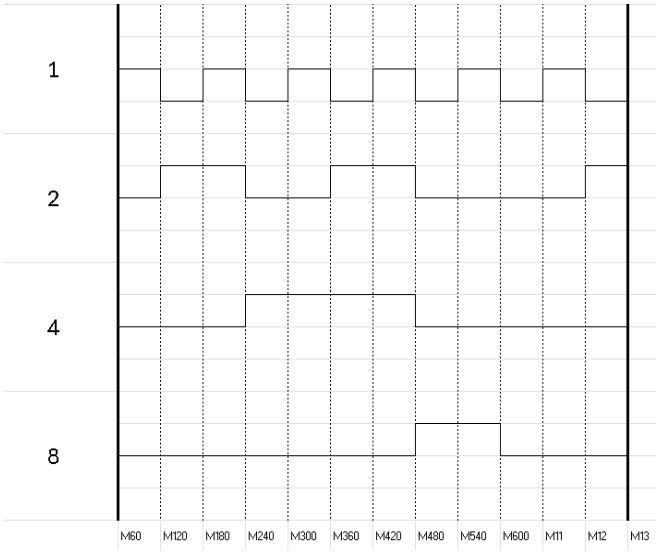

Fig. 2 Diagrama de tiempos para la generación de horas en formato binario.

Utilizar la lógica de los temporizadores en cascada junto con los diagramas de tiempos permite reducir la probabilidad de cometer errores, permitiendo la elección de los instantes en los que las salidas estarán activas para producir un número en los distintos Displays de manera más simple. Por esta razón

se diseña el diagrama de tiempos que se muestra en la Figura (2), el cual ayuda a identificar en que instante de tiempo se debe activar la entrada de los decodificadores BCD para generar su a salida la secuencia de números que van del número 1 al 12 y representar de este modo el curso de las horas.

La sección del diagrama en escalera diseñado como un contador de horas, se construye a partir de 6 temporizadores en cascada, cada uno de 10 minutos. De manera similar, se construye una sección del diagrama escalera que es la encargada de contar los minutos, para este caso, un temporizador de segundos llega hasta un valor de 60, mismo que permite registrar los minutos.

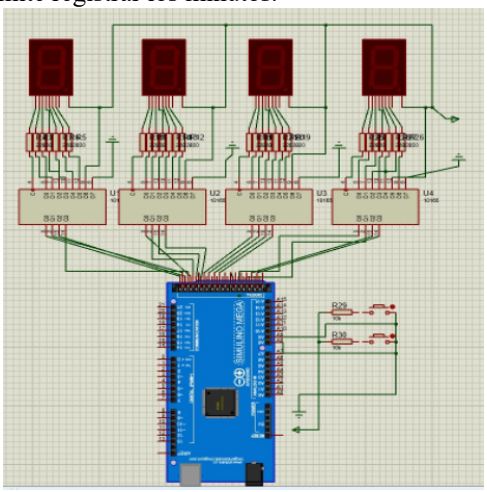

Fig. 3. Conexiones físicas entre Arduino Mega, decodificadores 74LS47 y Displays de 7 segmentos.

En la primer aproximación del prototipo de reloj digital se cuenta con una gran cantidad de conexiones entre el Arduino Mega, los decodificadores 74LS47, los Displays de 7 segmentos y los botones de arranque y paro, dichas conexiones que se encuentran representadas mediante el diagrama de la Figura (3) pueden provocar distintos problemas que van desde mal aspecto, hasta inducir y generar falsos contactos o cortos circuitos. Por este motivo los jumpers macho-macho y macho-hembra utilizados generalmente en prácticas de electrónica se sustituyeron por cable de red tipo UTP CAT5 cortado a medida. Posteriormente y con intención de brindarle un mejor aspecto al prototipo, se realizó un diseño de carcasa con dedos para su posterior corte laser sobre paneles de un material de fibra de madera conocido como MDF (Medium Density Fibreboard) para introducir la circuitería del reloj digital, tal como se muestra en la Figura (4).

Fig. 4 Diseño del programa para el corte de la carcasa.

A partir de la construcción de la carcasa hecha de MDF y manufacturada en corte laser, como se muestra en la Figura (5), es posible ocultar y proteger tanto las conexiones, como la circuitería del prototipo y brindarle al mismo una apariencia rustica de tipo *vintage*.

Fig. 5 Prototipo final, Reloj Digital.

El proyecto demostró la viabilidad de utilizar lenguaje escalera para la programación de un reloj digital en un entorno Arduino. El uso del lenguaje escalera incluye distintas ventajas, entre las cuales se destacan las que se presentan a continuación:

La estructura gráfica del lenguaje escalera facilitó el diseño modular y la identificación de errores en el programa. Utilizar Ladder Maker permitió una programación intuitiva y directa en el Arduino. El uso de bobinas temporizadas, en lugar de flip-flops controlado por señales de reloj como circuitos secuenciales, simplificó la implementación y permitió una mayor flexibilidad en el diseño. La solución es escalable para añadir más funciones, como una alarma o un calendario.

De forma complementaria, se detectaron distintas limitaciones que pueden corregirse en futuras aplicaciones, entre las cuales, es importante resaltar que la configuración de temporizadores precisos puede ser desafiante y requiere ajustes finos para mantener la precisión a largo plazo, incluso él requiere el uso de un módulo RTC (Real-Time Clock) como el DS3231 para garantizar una precisión horaria a largo plazo. Por tanto, la precisión del reloj está limitada por la estabilidad de los temporizadores del Arduino Mega, que pueden estar sujetos a variaciones. Por ejemplo, el Arduino Mega, utiliza un cristal de 16 MHz para su temporización interna y, por tanto, tiene una desviación de

aproximadamente ±30 segundos por día, o cerca de 1 segundo cada 3 minutos, que esta asociada a las limitaciones en la estabilidad del cristal, especialmente con variaciones de temperatura [7]. Como se ha mencionado, para mejorar la precisión en el Arduino Mega, es recomendable usar un módulo RTC, como el DS3231, que ofrece una precisión superior gracias a su oscilador de cuarzo compensado por temperatura, manteniendo una desviación de ±2 minutos por año, independientemente del microcontrolador utilizado.

A partir de las ventajas y limitaciones mencionadas anteriormente, es posible implementar un conjunto de mejoras, por ejemplo, resulta necesario integrar un módulo RTC (Real Time Clock)) para mejorar la precisión del tiempo, así como un conjunto de botones o una pantalla adicional para configurar la hora manualmente y más importante, es indispensable optimizar el circuito y la programación para reducir la demanda de recursos computacionales y el consumo energético.

IV. Conclusiones

El desarrollo de un reloj digital utilizando lenguaje escalera en un Arduino Mega fue exitoso, logrando los objetivos planteados de precisión y funcionalidad. La combinación de hardware y software seleccionados permitió una implementación eficaz, resaltando la versatilidad y potencia del lenguaje escalera en aplicaciones que pueden escalarse a nivel de control industrial. Además, la implementación de circuitos secuenciales mediante bobinas temporizadas en lugar de flip-flops demostró ser una alternativa viable y efectiva, pues en la implementación de este mismo reloj mediante flip-flops, es claro que a medida que se diseñan los circuitos para cada dígito (horas, minutos y segundos), el número de compuertas lógicas y flip-flops necesarios aumenta considerablemente, lo que ocupa mucho espacio en una protoboard, placa de prototipado e incluso e una PCB, aumentando en consecuencia la complejidad del diseño y montaje. Adicionalmente, los flip-flops y compuertas lógicas requieren un suministro constante de energía, especialmente en configuraciones complejas de múltiples dígitos y segmentos, lo cual eleva el consumo de energía y puede ser una limitación para sistemas alimentados por batería. De forma similar, al implementar un reloj digital con flip-flops, o algún otro circuito de naturaleza similar, la señal de reloj puede estar sujeta a ruidos y desviaciones, especialmente si se usa un circuito como el 555 TIMER IC, que puede tener variaciones en la frecuencia. Esto puede llevar a inexactitudes en el conteo del tiempo, generando atrasos o adelantos en el reloj con el tiempo.

En consecuencia, un reloj construido con compuertas lógicas y flip-flops, controlado por una señal de reloj como el 555 TIMER IC, tiene una precisión que depende de la estabilidad del oscilador. El 555 TIMER IC es un temporizador basado en un oscilador de condensador y resistencia (RC), cuya frecuencia puede variar significativamente con los cambios en temperatura y las variaciones de los componentes RC. Normalmente, un oscilador RC como el del UA55 puede tener una precisión entre ±1% y ±5%, lo que equivale a una desviación de varios minutos por día [8]. En términos prácticos, un error del 1% en la frecuencia podría generar una desviación de unos 14.4 minutos por día (aproximadamente 1 segundo cada minuto) y un error del 5% podría llevar a desviaciones de hasta 72 minutos por día, es decir, un reloj basado en el 555 TIMER IC y compuertas lógicas tiene una precisión limitada, normalmente en el orden de varios minutos de desviación diaria, lo que lo hace adecuado solo para aplicaciones donde la precisión de tiempo no es crítica. Para mejorar la precisión, se recomienda utilizar un oscilador de cuarzo en lugar de un oscilador RC, o un módulo RTC compensado por temperatura.

Finalmente y como fue observado en la presente implementación del reloj digital mediante diagramas escalera sobre Arduino Mega, las pruebas confirmaron la capacidad del sistema para mantener la precisión del tiempo en un formato de 12 horas, dado que el Arduino Mega cuenta con un reloj interno relativamente estable, incluso si la precisión estuviera afectada por problemas de desviación y ruido, el Arduino Mega puede sincronizarse con un módulo RTC (Real-Time Clock) como el DS3231, lo cual abre la puerta a futuras mejoras y aplicaciones más complejas utilizando metodologías similares.

Referencias

[1] Chinnery, D., Keutzer, K., Chinnery, D., & Keutzer, K. (2002). Reducing the Timing Overhead: Clock Skew, Register Overhead, and Latches vs. Flip-Flops. Closing the Gap Between ASIC & Custom: Tools and Techniques for High-Performance ASIC Design, 57-100.
[2] da Silva, C. C. A. F., Rêgo Segundo, A. K., & Lage, V. N. (2018). Low Cost Industrial Interface Design and Graphical Programming for Arduino. In Teaching and Learning in a Digital World: Proceedings of the 20th International Conference on Interactive Collaborative Learning–Volume 2 (pp. 37-42). Springer International Publishing.
[3] Geddes, M. (2017). Arduino Project Handbook, Volume 2: 25 Simple Electronics Projects for Beginners. No Starch Press.
[4] Wilcher, D. (2015). Arduino Electronics Blueprints. Packt Publishing Ltd.
[5] Waltech Systems. (n.d.). LadderMaker: Ladder logic software. Waltech Systems. https://waltech.com/cszcms/laddermaker.
[6] DIP, C. D. BCD TO 7-SEGMENT DECODER/DRIVER.
[7] Arduino, S. A. (2015). Arduino. Arduino LLC, 372.
[8] Mano, M. M. (2002). Digital design. Pearson Educación.