



JIDeTEV

Jornadas de Investigación y Desarrollo Tecnológico
Extensión, Vinculación y Muestra de la Producción



JIDeTEV- Año 2022 -ISSN 2591-4219

Temporizador programable a través de control remoto

Johann Mallmann Rodriguez ^{a*}, Joaquín J. Pelinski ^a, Juan P. Gross ^b, Guillermo A. Fernández ^b

^a Universidad Nacional de Misiones (UNaM). Facultad de Ingeniería. Misiones, Argentina.

^b Universidad Nacional de Misiones (UNaM)-CONICET. Facultad de Ingeniería. Instituto de Materiales de Misiones (IMAM). Grupo de Investigación y Desarrollo en Electrónica (GIDE). Misiones, Argentina.

johannmrodriguez@gmail.com, joaquinpelinski26@gmail.com, gross@fio.unam.edu.ar,
guillermo.fernandez@fio.unam.edu.ar

Resumen

Este trabajo consiste en el desarrollo de un sistema embebido que implementa el control de dos relés con accionamiento temporizado y es configurable mediante un control remoto infrarrojo. Se presenta el desarrollo del hardware y software del sistema mencionado, partiendo desde el diseño del circuito, pasando por la estructuración del programa ejecutado por el microcontrolador ATmega2560 utilizado, la realización del circuito impreso y finalizando en un sistema embebido totalmente representativo y funcional, que cumple con las especificaciones solicitadas. El resultado obtenido consiste en un sistema que posee dos relés, donde es posible activar/desactivar y configurar la temporización de cada uno de estos, en forma totalmente independiente, a través de un control remoto que envía señales infrarrojas. El sistema propuesto posee un display LCD que permita guiar al usuario en la configuración del mismo. Entre las conclusiones más importantes puede mencionarse la relevancia que tienen los sistemas embebidos en el día a día, facilitando la automatización de ciertas tareas; también se destaca la importancia en la interpretación de las hojas de datos de los componentes utilizados en el diseño de los sistemas embebidos y la articulación de los saberes requeridos para esto, los cuales son desarrollados en la asignatura donde surge el trabajo, como así también en otras previas que introducen y abordan conceptos necesarios para la realización de la actividad propuesta.

Palabras Clave – ATmega2560, Control Remoto, Microcontrolador, Programación, Relés, Temporización.

1. Introducción

El presente trabajo fue realizado en la asignatura Técnicas Digitales 2 de la carrera Ingeniería Electrónica, en la cual se estudian temas relacionados al desarrollo de sistemas embebidos basados en microcontroladores. El trabajo aquí presentado muestra los resultados de la actividad integradora final realizado en la asignatura mencionada, donde se requiere el desarrollo de un circuito electrónico con microcontrolador que disponga de dos relés con accionamientos programables a partir de un control remoto infrarrojo. Este último usado para la configuración y ajuste de los tiempos en que los relés deben estar activos (de 1 a 10 minutos), siendo estos tiempos configurables en forma totalmente independiente para cada relé del sistema. Por otra parte, la consigna de la actividad específica que el sistema a desarrollar cuenta con un accionamiento manual efectuado con un par de pulsadores para cada relé, los que serían responsables por empezar y finalizar la temporización del relé

correspondiente. Además, fue solicitado el uso de un display LCD para visualizar los tiempos y facilitar la configuración del sistema al usuario del mismo.

El circuito obtenido en este trabajo puede tener una amplia gama de aplicaciones, siendo ejemplos el control de motores, hornos, ventiladores, alarmas y demás aplicaciones relacionadas a la automatización de elementos semejantes. Las temporizaciones en la activación/desactivación de los relés pueden ser configuradas con precisión en el sistema desarrollado, debido a que es posible ajustar los tiempos en pasos de 1 segundo y de 10 segundos, ajuste fino y grueso, respectivamente. Además, la independencia de las temporizaciones proporciona una mayor dinámica y flexibilidad al sistema

A continuación, se presentan todos los pasos realizados para obtener un circuito totalmente funcional y de acuerdo a la consigna presentada en la actividad integradora propuesta. Para esto, inicialmente se describe el diseño del circuito y la programación de microcontrolador, luego los resultados donde se muestra el circuito impreso y la integración de los componentes en la placa desarrollada y finalmente las conclusiones obtenidas con el trabajo.

2. Diseño del hardware

Para el diseño del hardware, fueron consideradas cuatro partes, el circuito de control, la etapa de potencia, la fuente de alimentación y el control remoto. Estas partes se describen a continuación.

2.1. Circuito de control

La Fig. 1 representa el esquema del hardware desarrollado como solución a la problemática. El mismo está conformado por el circuito de control y la etapa de potencia.

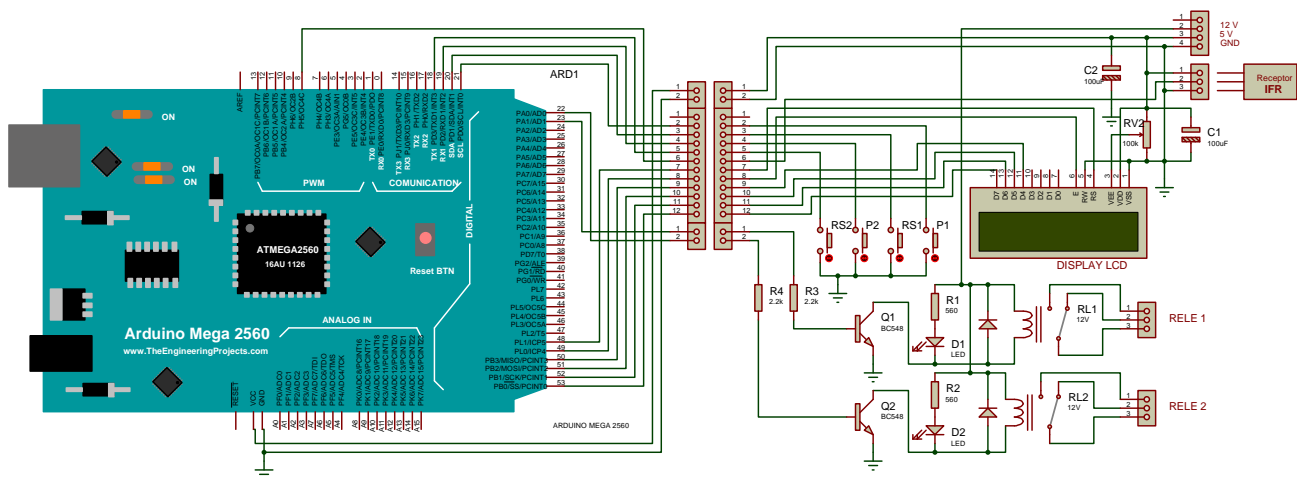


Fig. 1. Circuito principal del temporizador programable.

La etapa de potencia está constituida por los relés y sus transistores, junto a los leds indicadores; mientras que el circuito de control está conformado por la interfaz de usuario (display LCD y pulsadores) y la placa Arduino Mega que posee el microcontrolador de 8 bits ATmega2560 [1].

En el diseño del circuito de la Fig. 1, para comandar a los distintos elementos mediante el microcontrolador de la placa, líneas de entrada/salida de este dispositivo se configuraron de la siguiente forma:

- Entradas digitales: Receptor infrarrojo IFR para control remoto PH5; pulsadores: PD0 (P1), PD2 (P2), RS1 (PD1) y RS2 (PD3).
- Salidas digitales: Relé 1, PA1; Relé 2, PA2; display LCD: PL0 (E); PL1 (RS); PB0 (D7); PB1 (D6); PB2 (D5); PB3 (D4).

Los pulsadores P1 y P2 permiten iniciar la temporización de cada relé, mientras que RS1 y RS2 permiten resetear las mismas. Para el manejo de estos pulsadores, son utilizadas las interrupciones externas que dispone las entradas del puerto seleccionado.

Para visualizar la temporización en curso de los relés y guiar al usuario en la configuración de las mismas el circuito desarrollado posee un display LCD de dos filas por dieciséis caracteres, el cual es operado con datos de cuatro bits, haciendo uso de la librería “*LiquidCrystal*” para el control [2].

Todo el circuito de la Fig. 1 es energizado mediante una fuente de alimentación desarrollada para este trabajo la cual provee 5 V para energizar a la placa Arduino Mega y al display LCD y 12 V para alimentar a los relés del circuito.

2.2. Etapa de potencia del circuito

Para el diseño de la etapa de potencia se utilizan dos relés, permitiendo que el sistema propuesto pueda accionar diferentes tipos de cargas, que operan tanto con tensiones de CC como CA

Dado que la capacidad de corriente de las salidas digitales del ATmega2560 no es suficiente para comandar cada relé y su led indicador, las salidas accionan estos dispositivos mediante los transistores Q1 y Q2 de la Fig. 1, los cuales actúan como drivers de corriente. Considerando la corriente de colector y la tensión colector-emisor con que deben operar estos transistores, para los mismos se seleccionan el transistor bipolar NPN de propósitos generales 2N3904. Debido a la disponibilidad, para el relé se selecciona el modelo HJR-3FF-S-ZH junto a un led rojo difuso como indicador de su estado. En la Tabla 1 se resume las características de los componentes mencionados [3], [4].

El led indicador asociado a cada relé de la etapa de potencia, posee una resistencia limitadora de corriente la cual puede dimensionarse a través de las ecuaciones (1).

$$R_{led} = \frac{V_{CC} - V_{led} - V_{CEsat}}{I_{led}} \quad ; \quad P_{led} = I_{led}^2 R_{led} \quad (1)$$

Considerando las especificaciones de la Tabla 1 y las ecuaciones (1), la resistencia limitadora de corriente para cada led mencionado resulta con un valor comercial $R_{led} = 560 \Omega$ de 1/4 W.

En base a la corriente de colector obtenida mediante la ecuación (2), con las ecuaciones (3) puede dimensionarse la resistencia limitadora de corriente en la base de los transistores de la etapa de potencia.

$$I_c = I_{led} + I_{relé} \quad ; \quad I_b = 7 \cdot (I_c / h_{FEmin}) \quad (2)$$

$$R_b = \frac{V_{OH} - V_{BEsat}}{I_b} \quad ; \quad P_{R_b} = I_b^2 R_b \quad (3)$$

Siendo la mínima tensión en estado alto $V_{OH} = 4,1 \text{ V}$, para las salidas del microcontrolador que comandan el accionamiento de cada relé, la resistencia limitadora de corriente en la base de cada transistor resulta con un valor comercial $R_b = 2,7 \text{ k}\Omega$ de $1/4 \text{ W}$.

Tabla 1: Especificaciones técnicas de los componentes de la etapa de potencia.

Componente	Parámetro	Valor
Transistor 2N3904		
Tensión colector-emisor de saturación	V_{CEsat}	0,3 V
Tensión base-emisor de saturación	V_{BEsat}	0,95 V
Ganancia de corriente	h_{FEmin}	100
Led rojo difuso		
Corriente principal	I_{led}	20 mA
Caída de tensión en directa	V_{led}	2 V
Relé HJR-3FF-S-ZH		
Corriente nominal de la bobina	$I_{relé}$	30 mA
Voltaje nominal de la bobina	$V_{relé}$	12 V

Seguidamente es presentado el diseño de la fuente de alimentación encargada de energizar el circuito de Fig. 1 a partir de la red eléctrica.

2.3. Fuente de alimentación

La fuente propuesta para energizar al circuito, es considerada necesaria dado que el sistema debe tener una única fuente de alimentación que sea independiente de la conexión USB utilizada para programar al microcontrolador. Siendo así, para el display LCD, la placa Arduino y el receptor infrarrojo IFR la fuente diseñada provee 5 V, mientras que para los relés entrega una tensión de 12V. La solución propuesta es la presentada en la Fig. 2.

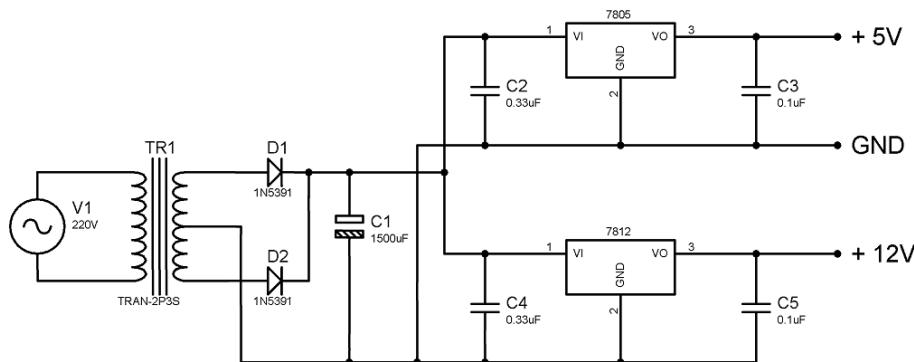


Fig. 2. Circuito de la fuente de alimentación.

Midiendo el consumo de corriente que debe abastecer cada tensión de la fuente de alimentación, se obtiene 57 mA para la tensión de 5 V y 90 mA para la tensión de 12 V. Como en ambos casos el consumo es menor a 1 A, con el fin de que la fuente proporcione voltajes de salida regulados, en el

circuito de la misma son utilizados los reguladores de tensión LM7805 y LM7812 respectivamente, los cuales necesitan en su entrada una tensión rectificada y filtrada de por lo menos 2V mayor a la tensión de salida [5]. Atendiendo a esto, se adopta $V_{in} = 18$ V para la tensión de entrada a ambos reguladores, la cual está relacionada con la tensión eficaz del secundario del transformador V_{ef} mediante la ecuación (4).

$$V_{in} = \sqrt{2} \cdot V_{ef} - V_D \quad (4)$$

Para el cálculo del capacitor de filtrado $C1$, es considerado un factor de *ripple* del $F_r = 5\%$. Por lo tanto, la tensión de *ripple* V_r puede determinarse con la ecuación (5).

$$F_r = \frac{V_r}{V_{in}} \times 100 \quad (5)$$

A partir de la tensión de *ripple* V_r y la corriente total I_{sal} que debe suministrar el rectificador y el filtro, el valor del capacitor $C1$ puede determinarse con la ecuación (6).

$$C1 = \frac{I_{sal}}{2 \cdot \sqrt{2} \cdot f \cdot V_r} \quad (6)$$

Considerando un factor de seguridad de 1,2 en la corriente que debe suministrar cada tensión de alimentación de la fuente y sumando las mismas, resulta $I_{sal} = 180$ mA. A partir del factor de *ripple* considerado, se obtiene $V_r = 0,9$ V. Con estos valores, el valor comercial para el capacitor del filtro de la fuente de alimentación resulta $C1 = 1500 \mu\text{F} / 35$ V.

En la entrada y salida de los reguladores de tensión, los capacitores $C2$, $C3$, $C4$, $C5$ actúan como filtros para atenuar ruido, los valores adoptados son los recomendados por el fabricante en la hoja de datos, siendo $C2 = 0,33 \mu\text{F} / 35$ V; $C4 = 0,33 \mu\text{F} / 35$ V, $C3 = 0,1 \mu\text{F} / 50$ V, $C5 = 0,1 \mu\text{F} / 50$ V.

Considerando que los diodos utilizados en el rectificador de la fuente de alimentación son del tipo 1N5391, con una caída de tensión $V_D = 1,3$ V, a partir de la ecuación (4) la tensión eficaz de salida del transformador resulta $V_{ef} = 15$ V. Por lo cual, es seleccionado un transformador de 220 V / (16+16) V / 500 mA.

2.4. Receptor infrarrojo y control remoto

Para configurar la temporización de los relés y comandar el accionamiento de los mismos, se ha seleccionado el receptor infrarrojo AX-1838HS el cual puede operar con las señales emitidas por el control remoto de la Fig. 3. El receptor mencionado posee un fotodetector con preamplificador y un filtro para modulación de pulsos codificados [6].



Fig. 3. (a) Receptor infrarrojo AX-1838HS; (b) Control remoto.

A través del control remoto mostrado en la Fig. 3b, el usuario puede definir las temporizaciones de los relés. Esto puede efectuarse de dos formas distintas. La primera forma corresponde a cuando el sistema es energizado; en esta etapa el usuario puede ingresar los minutos y segundos para cada relé separadamente y luego de eso, al ingresar los segundos del último relé, la temporización comienza a correr y el display muestra la cuenta regresiva para ambos relés. La segunda forma de definir las temporizaciones, es durante el proceso de temporización de cualquiera de los relés, en esta situación puede reconfigurarse cada tiempo haciendo uso de las teclas 1 (relé 1) y 2 (relé 2), según cual se desee; en esta etapa la configuración se basa simplemente ajustar el tiempo original en pasos de 1 ó 10 segundos, restando (con teclas -, <<) o sumando (con teclas +, >>) al tiempo original. Las funciones de cada tecla se describen en la Tabla 2.

Tabla 2. Función individual de las teclas del control remoto.

Tecla	Función
CH-	Apaga/enciende el relé 1 y resetea el tiempo.
CH+	Apaga/enciende el relé 2 y resetea el tiempo.
CH	Apaga/enciende ambos relés (únicamente si ambos ya fueron configurados)
1	Activa configuración del relé 1.
2	Activa configuración del relé 2.
0-9	Valores numéricos para los tiempos.
Play/Pause	Salida de la configuración de temporización.
-	Ajuste fino – disminuye el tiempo en 1s.
+	Ajuste fino – aumenta el tiempo en 1s.
<<	Ajuste grueso – disminuye el tiempo en 10s.
>>	Ajuste grueso – aumenta el tiempo en 10s.

A seguir se detalla la programación del microcontrolador. Explicando la lógica del programa y la manera en que es posible generar la temporización de una manera precisa.

3. Diseño del software

El programa desarrollado para el microcontrolador ATmega2560 se realizó utilizando el Entorno Integrado de Desarrollo (IDE) de Arduino. El programa utiliza dos librerías externas, una para manejar el display LCD y otra para leer el receptor IFR [7], [8]. A continuación, se describen las distintas etapas del programa que ejecuta el microcontrolador mencionado.

3.1. Operación del programa del microcontrolador

Inicialmente, en el programa son definidas las variables globales para representar a cada tecla del control remoto, mediante un número hexadecimal constante.

Luego, el programa consta de dos partes, una no repetitiva denominada “*setup*” y otra parte repetitiva denominada “*loop*”. En la parte no repetitiva del programa están configurados los módulos temporizadores (*timers*) que posee el microcontrolador, usándose el *Timer 1* y el *Timer 3* [1], [9]; acompañando a esto la configuración de las interrupciones externas (INT0 a INT3) asociadas a los pulsadores de la placa (P1, P2, RS1 y RS2), la inicialización del display LCD con un mensaje de aviso y la configuración de los puertos del microcontrolador utilizados por los periféricos.

Antes de la etapa repetitiva, en el programa posee la declaración y desarrollo de las rutinas de servicio a las interrupciones utilizadas. Estas rutinas corresponden a un evento incitado por una señal en algún puerto de interrupción, dada por alguno de los pulsadores, esto quiere decir que, cuando existe un uno lógico en un alguno de estos puertos, se da lugar a un subproceso del programa, controlado por lo temporizadores dichos anteriormente y explicados en la sección 3.2.

La etapa repetitiva del programa está dedicada a la lectura del receptor IRF que cuando sensibilizado por alguna señal emitida desde el control remoto, da lugar a la función correspondiente a la tecla presionada en disco control. En estas funciones, estructuras de control y de repetición son predominantes, ya que determinan el flujo del programa en base a las variables.

Existen siete funciones. La primera es la responsable de verificar si alguna tecla del control remoto fue presionada y así retornar el valor hexadecimal al flujo principal. Luego existe la función que es responsable de escribir en el display, en base a la situación en que se encuentran los relés, encendido y tiempo restante o apagado, para eso se realiza una consulta a las variables que representan cada uno de los relés y se realiza el llamado a las funciones que controlan los tiempos. Además, existen dos funciones que controlan a cada relé individualmente, pero simultáneamente, utilizando un *Timer* cada una. La estructura responsable de definir el tiempo inicial, es llamada cuando se inicia el programa por primera vez, en la cual basta ingresar los minutos y los segundos por el control. Sumada a estas, existe una función para convertir los minutos en segundos y sumarlos con los segundos restantes, esto quiere decir que, el microcontrolador trabaja con los temporizadores en segundos, siendo así, es lógico que se emplee segundos como medida, sin embargo, para el usuario es más conveniente definir el tiempo en minutos y segundos, de esto surge la importancia de esta estructura. Finalmente, está la función responsable por escribir en el display, que recibe como argumentos dos buffers modificados en el programa de acuerdo con el estado de los relés. Todas estas funciones se repiten más de una vez en cada ciclo *loop* del programa.

En conclusión, el programa interpreta la tecla presionada y ejecuta una secuencia de comandos, solicitando que el usuario interactúe mediante el control cuando es necesario. El display sirve como guía visual para la configuración del sistema y los pulsadores como dispositivos de interrupción en caso de emergencia.

3.2. Configuración de los timers del microcontrolador

El programa desarrollado utiliza dos *timers* del microcontrolador. El *Timer 1* (de 16 bits) es el responsable de generar una base de tiempo de 1 segundo para controlar y mantener los relés activados.

Este timer es configurado para operar en modo *free-running* con valor inicial $VPC1$ y un prescaler N . El valor de temporización con el mismo, puede calcularse mediante la ecuación (7).

$$T = \frac{(65535 - VPC1) \cdot N}{f_{osc}} \quad (7)$$

Para lograr una temporización $T = 1$ s, considerando que la frecuencia del cristal oscilador del microcontrolador es $f_{osc} = 16$ MHz y que se adopta un *prescaler* $N = 1024$, el valor inicial para el Timer 1 resulta $VPC1 = 49910$. Con este valor, este *timer* genera interrupciones cada 1 s. Cabe aclarar que, se determinó como caso ideal 1 segundo debido a que el control era más sencillo y preciso convirtiendo todo el tiempo en segundos. De esta manera, los contadores de cada relé almacenan los valores en segundos.

Un segundo *timer* fue empleado en la parte de ajuste del tiempo. Cuando ya existe un tiempo inicial configurado, la opción es sumar o restar 1 o 10 segundo. De esta manera al mantener pulsada una tecla, lo correcto sería que se reflejara un aumento más rápido del valor. Como eso resultaría imposible al tener una actualización a cada 1 segundo debido a la interrupción del Timer 1, se configuró otro temporizador responsable por actualizar los valores a cada 0,2 segundos y solo puede ser accionado por un relé, sin afectar la temporización del otro, y en un único “lugar” del programa, el cual corresponde al ajuste de tiempo del relé en cuestión.

A partir de lo mencionado, se utiliza el Timer 3 (de 16 bits) que también es configurado para operar en modo *free-running*. Siendo $T = 0,2$ s y $N = 1024$, a partir de la ecuación (7), para este *timer* el valor inicial es $VPC3 = 62410$.

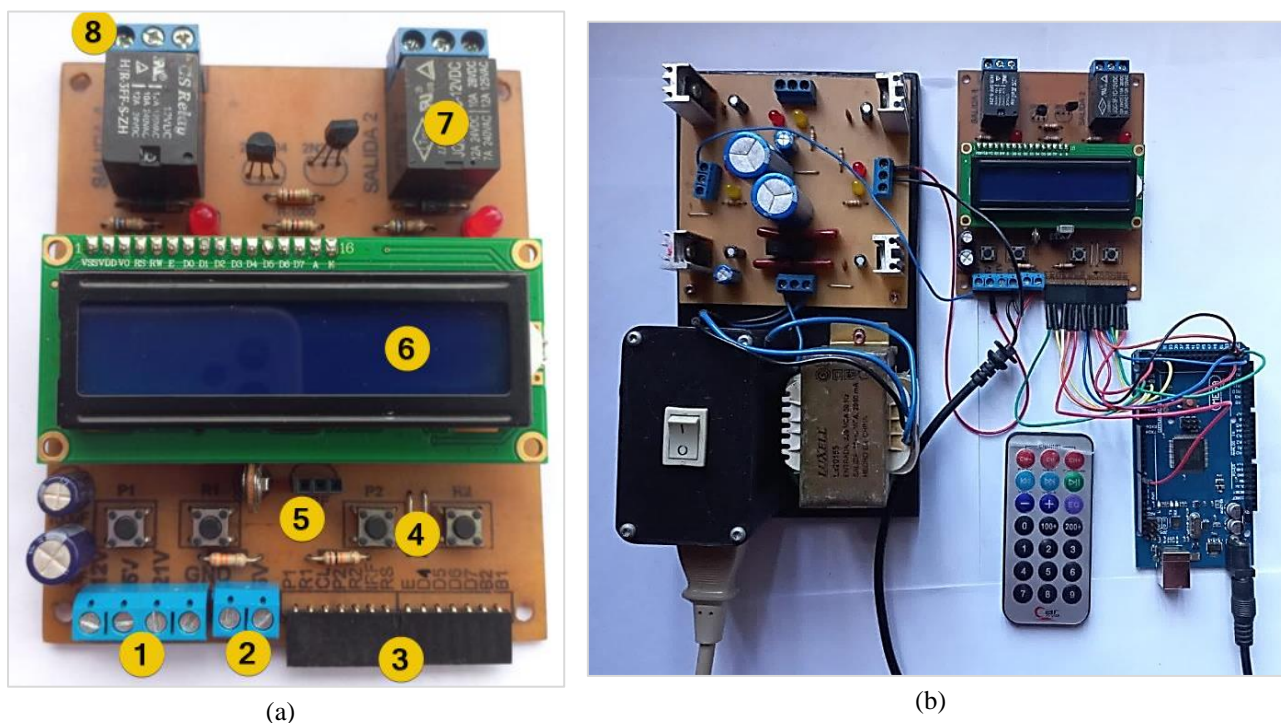


Fig. 4. Prototipo de sistema desarrollado: (a) Hardware finalizado; (b) Sistema con todos sus componentes.

4. Resultados

Para depurar y comprobar operación del programa explicado en la sección anterior, se armó el circuito de la Fig. 1 en un *protoboard*. Una vez que el programa proporcionó los resultados deseados, se procedió al diseño del circuito impreso, el montaje y el soldado de los componentes, obteniéndose la placa mostrada en la Fig. 4a. En esta figura puede observarse los componentes presentados en el esquema inicial del hardware (Fig. 1). El elemento 1 corresponde a la bornera para conectar las salidas de la fuente de alimentación (12 V, 5 V y GND) que proporciona las tensiones de alimentación requerida en el circuito. El elemento 2 constituye una bornera de salida, a través de la cual se obtiene la tensión de alimentación para la placa Arduino Mega. El elemento 3 es una tira de pines hembra que permite conectar al circuito desarrollado con la placa Arduino Mega; posee las conexiones del receptor infrarrojo, el display LCD, los pulsadores y relés. Los pulsadores representados por 4 corresponden a P1, P2, RS1 y RS2 de la Fig. 1. El conector identificado con número 5, permite la conexión del receptor infrarrojo, necesario para detectar el accionamiento de las teclas del control remoto. El elemento 6 es el display LCD a través del cual el usuario puede guiarse en la configuración de las temporizaciones. Los relés que posee el sistema están indicados con 7, estos permiten el accionamiento temporizado de los dispositivos que lo requieran. Finalmente, con 8 están identificadas las borneras que constituyen los contactos NA, NC y C de cada relé, a través de las cuales pueden conectarse los dispositivos externos que se desean controlar.

Tabla 3: Lista componentes del sistema y precios.

Nombre del ítem	Cant. (unid.)	Precio unit. (\$)	Precio x componente (\$)
Arduino MEGA	1	7000	7000
Display LCD 16x2	1	1000	1000
Kit Control Remoto (receptor IR + control)	1	645	645
Relé	2	400	800
Transistor BC3904	2	30	60
LED Rojo 5mm	2	25	50
Pulsadores	4	25	100
Resistencias 1k Ω -5%-0,25W	2	15	30
Resistencias 580 Ω -5%-0,25W	2	15	30
Preset 100 k Ω	1	60	60
Capacitor 100 μ F/35V	2	30	60
Montaje PCB	1	1500	1500
Transformador (220V; 16+16; 1.5A)	1	2000	2000
Capacitor 3900 μ F/16V	1	250	250
Capacitor 0.33 μ F/16V	2	25	50
Capacitor 0.1 μ F/16V	2	20	40
Diodo 1N5391	4	15	60
Regulador 7805	1	220	220
Regulador 7812	1	220	220
Montaje PDC	1	1600	1600
Precio Final (al 05/08/22)			\$15.775,00

Una vez obtenida la placa de la Fig. 4a, pudo verificarse el correcto funcionamiento del circuito mediante la comprobación de las temporizaciones configuradas en cada relé individualmente. El tiempo fue cronometrado y no se encontraron fallas. Gracias a los resultados obtenidos, puede afirmarse que el sistema desarrollado funciona de acuerdo a lo solicitado en las especificaciones requeridas, pudiendo operarse el mismo una manera sencilla para cualquier persona.

5. Conclusiones

El resultado final obtenido en este trabajo, mostrado en la Fig. 4b, se logró gracias a la integración de saberes desarrollados en otras asignaturas de la carrera. Conceptos presentados en materias anteriores resultaron de suma importancia para obtener los resultados presentados. Asignaturas como Informática y Computación, aportaron conocimientos sobre el desarrollo de algoritmos orientados a una tarea específica; asignaturas como Dispositivos Electrónicos y Electrónica Analógica permitieron brindar conocimientos acerca de transistores, relés y otros componentes, como así también el diseño de fuentes de alimentación como la desarrollada en el presente proyecto; mientras que la asignatura Técnicas Digitales 1 aportó los conceptos relacionados a los elementos del circuito y de cómo interpretar correctamente las hojas de datos de los mismos.

En cuanto a la asignatura Técnicas Digitales 2, la realización de este trabajo permitió profundizar el aprendizaje sobre temáticas relacionadas con el diseño y construcción de sistemas embebidos en base a microcontroladores y los recursos que poseen estos dispositivos, como ser puertos de entrada/salida y temporizadores internos (*timers*). Siendo estos últimos muy importantes para el desarrollo de algoritmos que involucran un determinado tiempo de ejecución, posibilitando el control dinámico del funcionamiento del sistema.

Con el objetivo de finalizar el presente artículo, es posible concluir que se han cumplido los objetivos propuestos, permitiendo obtener el prototipo de un temporizador programable mediante control remoto, sencillo y fácil de operar, el cual fue obtenido gracias a la integración de los saberes propios de la asignatura en la que surge la actividad, como así también de otros adquiridos en asignaturas previas.

Referencias

- [1] Atmel. ATmega640/V-1280/V-1281/V-2560/V-2561/V. Disponible en: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf
- [2] Librería LiquidCrystal para display LCD. Disponible en: <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>
- [3] Hoja de datos del Transistor BC3904. Disponible en: <https://pdf1.alldatasheet.com/datasheet-pdf/view/11471/ONSEMI/2N3906.html>
- [4] Hoja de datos del relé HJR-3FF-S-ZH. Disponible en: <https://www.maritex.com.pl/product/attachment/86515/HJR-3FF.pdf>
- [5] Hoja de datos del regulador LM78xx. Disponible en: <https://pdf1.alldatasheet.es/datasheet-pdf/view/105712/FAIRCHILD/LM7812.html>
- [6] Hoja de datos del receptor infrarrojo AX-1838HS. Disponible en: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132539/ETC2/AX-1838HS.html>
- [7] Datasheet Display LCD 16x2. Disponible en: <https://html.alldatasheet.com/html-pdf/251969/VISHAY/LCD-016M002B/221/1/LCD-016M002B.html>
- [8] Librería de Sensor Infrarrojo IFR, disponible en: <https://github.com/Arduino-IRremote/Arduino-IRremote>
- [9] Apuntes y presentaciones de la asignatura de Técnicas Digitales 2, 2021.