



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



JIDeTEV- Año 2021 -ISSN 2591-4219

Temporizador de Intervalos Programable

Joaquín A. Gonzalez ^{a,*}, Sergio A. Pacheco ^a, Juan P. Gross ^{b,c}, Guillermo A. Fernández ^{b,c}

^a *Universidad Nacional de Misiones, Facultad de Ingeniería. Oberá, Misiones, Argentina.*

^b *Universidad Nacional de Misiones, Facultad de Ingeniería, Instituto de Materiales de Misiones (IMAM), Misiones, Argentina.*

^c *ETCOLAB, GIDE, Juan Manuel de Rosas 325, Oberá, Misiones, Argentina.*

gonzalez.joaquin.ev@gmail.com; sergioarielpacheco@gmail.com; gross@fio.unam.edu.ar; fernandez@fio.unam.edu.ar

Resumen

El presente artículo describe las actividades desarrolladas en el Trabajo Integrador de la asignatura Técnicas Digitales 2, correspondiente al cuarto año de la carrera Ingeniería Electrónica. El trabajo consistió en el desarrollo de un Temporizador Digital Programable, capaz de activar dos relés de manera independiente según el tiempo que el usuario ha establecido para cada uno a través de un teclado conformado por doce pulsadores. El circuito está basado en el microcontrolador ATmega 2560 de la plataforma Arduino Mega. Primeramente, en el desarrollo de este trabajo se planteó una posible solución, diseñándose el esquema eléctrico del circuito, distintos diagramas de flujos y el programa en lenguaje C. Luego se realizaron pruebas de funcionamiento con simulación, comprobándose la operación correcta, tanto del circuito como del programa desarrollado para el microcontrolador. Seguidamente el circuito fue armado en un protoboard, donde se efectuaron los ajustes necesarios para alcanzar los resultados solicitados en la consigna del trabajo. Esto permitió comprobar el correcto funcionamiento del sistema propuesto, verificándose la temporización programada con el temporizador de un celular smartphone. Con esto puede concluirse que el trabajo resultó ser una buena experiencia para saber resolver un problema tecnológico donde existen algunas especificaciones técnicas solamente, siendo la solución pensada, elegida y desarrollada por el profesional de acuerdo a sus conocimientos y experiencia.

Palabras Clave – *Temporizador; programable; microcontrolador; automatización; lenguaje C; programación.*

1. Introducción

En el ámbito de los circuitos digitales, un Temporizador Programable (TP) es un contador que dispara un evento cuando alcanza la cuenta programada. Estos temporizadores pueden ser one-shot o periódicos. Los one-shot disparan el evento solo una vez y detienen la cuenta. Los temporizadores

*Autor en correspondencia.

periódicos disparan un evento cada vez que alcanzan determinado valor, produciéndose así una serie de eventos en intervalos de tiempo regulares.

El uso de los Temporizadores Programables puede aportar dos ventajas importantes donde se los aplica [1]:

- Control semiautomático sobre el tiempo de funcionamiento de dispositivos que operan con energía eléctrica.
- Reducción del consumo de energía eléctrica y eliminación del consumo en stand-by de algunos de los dispositivos mencionados.

El Temporizador Programable desarrollado en este trabajo es del tipo one-shot y el evento que produce cuando se alcanza la cuenta programada es el accionamiento de un relé, el cual puede utilizarse para abrir o cerrar un circuito eléctrico. El temporizador desarrollado en este trabajo funciona con un tiempo programable por el usuario, que puede ser en segundos, minutos u horas, siendo los valores máximos y mínimos de 3 horas y 10 segundos respectivamente.

La Fig. 1 muestra las distintas etapas del circuito construido. La programación del tiempo de funcionamiento se realiza mediante un teclado conformado por 12 pulsadores. La temporización programada es de conteo regresivo durante su transcurso. El usuario inicia el conteo de forma manual, accionando un pulsador del teclado, que también permite detener la cuenta antes de que el tiempo programado llegue a cero. El circuito cuenta con un Display LCD para mostrar el tiempo programado del TP, así como también el tiempo restante para la finalización de la temporización.

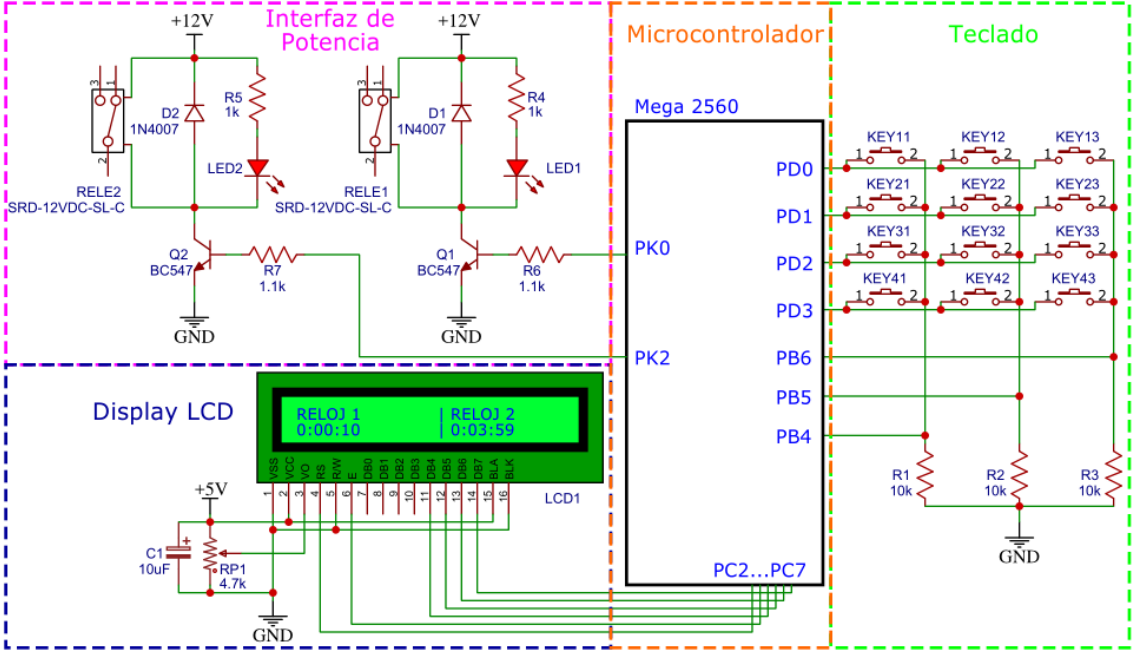


Fig. 1. Circuito del Temporizador de Intervalos Programables.

2. Diseño del Temporizador Programable

El diseño del Temporizador Programable se dividió en cuatro partes. Inicialmente se procedió al desarrollo del código del programa en lenguaje C. Este programa contiene la secuencia de instrucciones que debe procesar el microcontrolador para llevar a cabo el correcto funcionamiento del TP. En el diseño del programa se establece la manera en la que el microcontrolador recibirá la información proveniente del teclado, cómo se visualizará el funcionamiento del TP a través del display LCD, la lógica detrás del encendido/apagado de los relés y la administración de los tiempos de cada temporización. Luego se estableció la configuración del teclado, el cual resultó en un conjunto de pulsadores y resistencias. Seguidamente se diseñó la etapa de potencia que hace de interfaz entre los relés y el microcontrolador, comunicando las señales provenientes de este último a los respectivos relés que posee el circuito. Finalmente, se estableció el funcionamiento de la parte del programa que permite la visualización de información relevante en el display LCD.

2.1. Programa para el microcontrolador

El microcontrolador utilizado es el ATmega 2560 contenido en la placa Arduino Mega [2]. Esta placa cuenta con pines de entrada/salida digitales y analógicas, pines de comunicación serial y salidas PWM. Una herramienta útil para este proyecto, es el módulo interno llamado “Timer” que posee el microcontrolador del Arduino Mega [2]. Este módulo permite temporizar una determinada cantidad de tiempo de forma independiente a las tareas llevadas a cabo por el programa principal que ejecuta el microcontrolador [3].

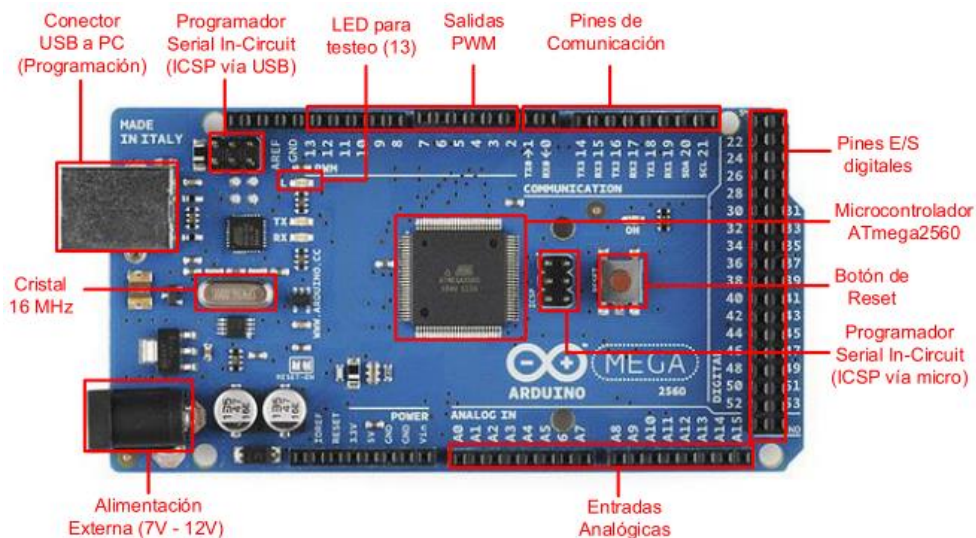


Fig. 2. Placa de desarrollo Arduino Mega [3].

En la Fig. 2 se observa la placa Arduino Mega con la distribución de sus pines y entrada de alimentación [3], la cual puede hacerse a través de una PC mediante un cable USB, o también con una fuente de alimentación externa (7V a 12V) [2].

Para controlar la operación de los relés que posee el Temporizador Programable, se utilizan los pines digitales PK0 y PK2 (configurados como salidas) del microcontrolador. Cuando se desea encender un relé, el programa realizado envía una señal correspondiente a un 1 lógico (entre 4.2V y 5V) y para apagarlo el mismo envía un 0 lógico (entre 0V y 0.9V) en los pines mencionados. La operación del teclado usa cuatro salidas digitales (PD0, PD1, PD2 y PD3) y tres entradas digitales (PB4, PB5 y PB6) del microcontrolador. Para la comunicación con el Display son utilizadas seis salidas digitales (desde PC2 a PC7).

La codificación del programa en lenguaje C se realizó en el software Atmel Studio [4]. Este software permite escribir el código del programa, realizar la depuración del mismo y cargarlo a la placa Arduino Mega a través de un cable USB. En el código C se programan las funciones de cada pulsador del teclado, así como la propia temporización, las condiciones para el encendido/apagado de los distintos relés, y la visualización de las temporizaciones de cada uno en el Display.

El microcontrolador utilizado posee cinco temporizadores o timers, los cuales pueden tener contadores de 8 bits o de 16 bits [5]. La diferencia entre el número de bits es el valor máximo de tiempo que pueden contar estos recursos, siendo de 16,384 ms para el timer de 8 bits y de 4,194304 segundos para los timers de 16 bits. Estos timers sirven, entre otras cosas, para manejar funciones de tiempo y generar señales periódicas según la configuración de los mismos. En el desarrollo del programa de este trabajo se utilizan dos timers de 16 bits, los módulos Timer1 y Timer3. Cada uno se configura en “Modo normal” con una cuenta de 1 segundo. A continuación, se explica el funcionamiento de los timers en este modo.

En modo normal cada timer cuenta en forma ascendente los pulsos de una señal de reloj. Este conteo sucede en simultáneo con los procesos llevados a cabo por el microcontrolador. El valor del conteo se guarda en un registro interno del microcontrolador llamado TCNTn. Para cada registro, la letra “n” representa el número del timer, teniendo entonces para ambos timers, los registros TCNT1 y TCNT3. Esta nomenclatura se utiliza para cada registro mencionado en adelante. Cuando el contador del timer alcanza el valor máximo, 0xFFFF (en formato decimal 65535), el mismo se resetea a 0x0000 como lo indica la Fig. 3(a). Cuando esto sucede, un registro interno del Timer, llamado TOVn, pasa de 0 a 1 [1] y el microcontrolador deja la rutina principal de instrucciones y realiza un salto para ejecutar una rutina de servicio a la interrupción (RSI), siempre que la interrupción se encuentre activada. Cuando se completa la RSI, la CPU del microcontrolador retorna a la rutina principal ejecutando la siguiente instrucción a la que se dejó para atender a la interrupción.

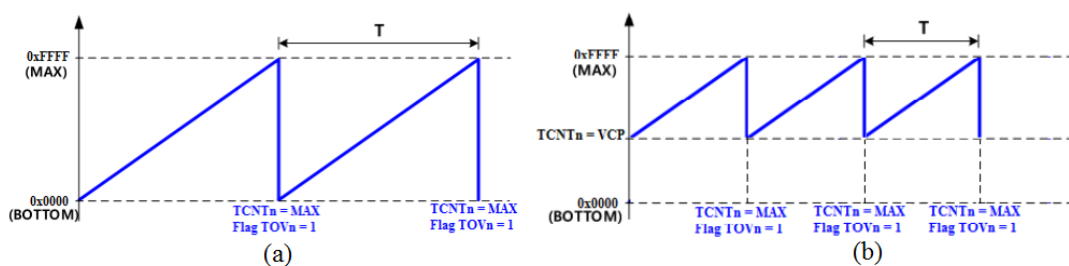


Fig. 3. Funcionamiento del módulo Timer en modo normal.

A partir de lo mencionado, con cada timer usado en el programa del microcontrolador puede obtenerse un tiempo máximo T levemente superior a los cuatro segundos. En el TP diseñado cada timer alcanza una temporización de 1 segundo. Para llegar a este valor se ajusta el número desde el cual el Timer empieza la cuenta, llamado este valor de precarga VCP . De esta forma, los timers usados operan como se observa en la Fig. 3(b). De acuerdo al tiempo deseado, VCP puede obtenerse de la siguiente ecuación [1].

$$T = \frac{(65536 - VCP)N}{FCLK_{ATmega2560}} \quad (1)$$

Donde $FCLK_{ATmega2560}$ corresponde a la frecuencia de la señal de reloj con que opera el microcontrolador, en este caso es de 16 MHz; N es el valor de división de frecuencia (o de prescala) que se afecta a la señal de reloj mencionada y puede adoptar los valores 1, 8, 64, 256 ó 1024. Para obtener una temporización de $T = 1$ s, con $FCLK_{ATmega2560} = 16$ MHz y $N = 1024$, a través de la Ec. (1) se obtiene un valor de precarga $VCP = 49911_{10}$ ($C2F7_{16}$) utilizado para el inicio de la cuenta de cada timer.

Lo siguiente es programar los momentos de inicio y finalización de los Timer, así como la acción que realizará el microcontrolador cada vez que algún timer finalice su cuenta (RSI).

Una vez que el usuario inicia la temporización de alguno de los dos relés a través del pulsador correspondiente, el programa del microcontrolador activa la función Timer energizando el respectivo relé. Cada vez que el timer correspondiente realiza la cuenta de un segundo, el tiempo de funcionamiento programado a través del teclado, es decrementado. Cuando este tiempo llega a cero, el programa da la orden de desenergizar el relé y también desactiva el timer correspondiente. Esto puede apreciarse en el diagrama de la Fig. 4, el cual ilustra cómo el programa del microcontrolador decrementa el tiempo cuando un timer correspondiente se encuentra activado y finaliza la cuenta de 1 segundo.

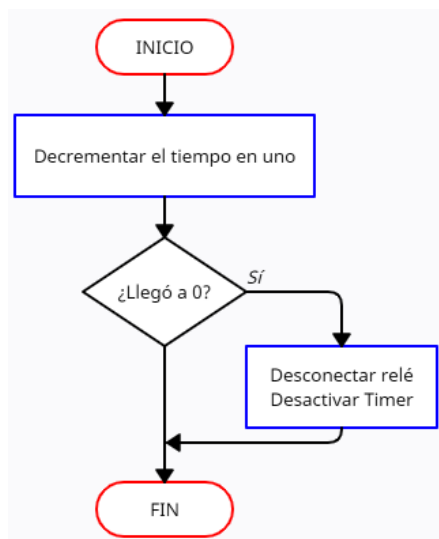


Fig. 4. Diagrama de flujo de las acciones realizadas por cada interrupción del Timer.

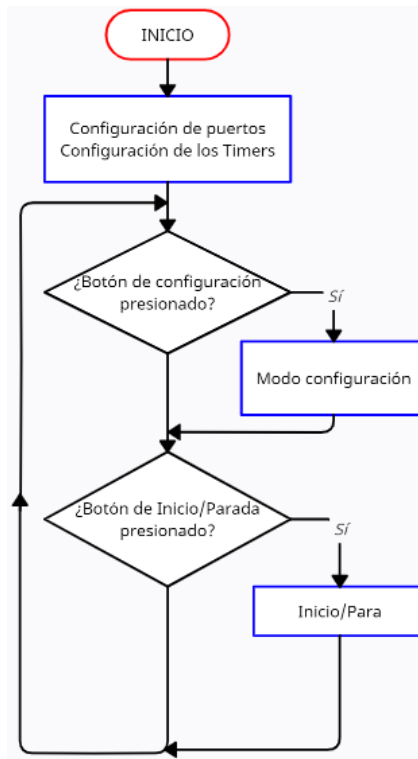


Fig. 5. Diagrama de flujos del programa principal.

La Fig. 5 muestra el diagrama de flujo general del programa ejecutado por el microcontrolador. La manera de realizar la configuración del tiempo de cada temporizador se explica en la siguiente sección.

2.2. Operación del teclado

La Fig. 6 muestra las funciones de los pulsadores del teclado matricial usado como interfaz de usuario para poder configurar las temporizaciones en el TP.

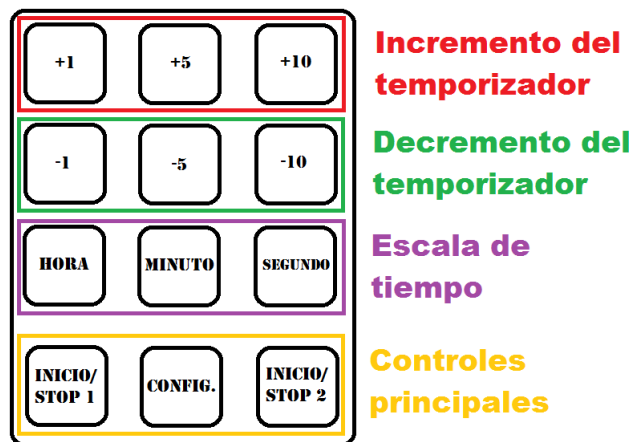


Fig. 6. Funcionalidades del teclado.

La Fig. 6 muestra el teclado con los pulsadores separados según su función. Las teclas principales son las del grupo llamado **Controles principales**. Estas sirven para poner en marcha o detener las temporizaciones de los relés (pulsadores Inicio/Stop) y para entrar en el modo de configuración (pulsador Config) donde el usuario puede configurar los tiempos de funcionamiento de cada relé. Dentro del modo configuración se usan los nueve pulsadores restantes que posee el teclado en las filas superiores. Los pulsadores de **Escala de tiempo** permiten elegir la unidad en la que se configurará la temporización deseada, pudiendo ser esta en segundo, minuto u hora. Los grupos **Incremento y Decremento del temporizador** contienen los pulsadores para aumentar o disminuir el tiempo configurado en valores por unidad. Los pulsadores de la primera fila del teclado sirven para aumentar el tiempo de la temporización a programar en 1, 5 o 10 valores por unidad. Con los pulsadores de la segunda fila puede disminuirse el tiempo en 1, 5 o 10 valores por unidad.

Como se mencionó anteriormente, la manera de conectar el teclado al microcontrolador es mediante cuatro pines configurados como salida (PD0 a PD3) y tres pines configurados como entradas (PB4 a PB6). Esto puede apreciarse en la Fig. 7, donde los pines de salida (verde) comandan las filas del teclado, mientras que los pines de entrada (rosa) están conectados a las columnas del mismo. Al teclado de la Fig. 7 se lo conoce como “Teclado matricial” [6]. A continuación, se explica el funcionamiento del mismo en conjunto con el microcontrolador. Cabe mencionar que en ningún momento se energizan las cuatro filas al mismo tiempo, sino que una por vez.

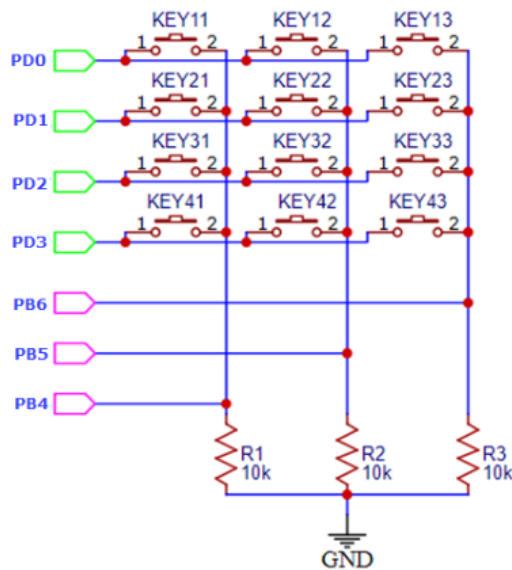


Fig. 7. Conexión de los pulsadores.

La Fig. 8 muestra el procedimiento para la detección de un pulsador presionado que realiza el programa del microcontrolador. En la lectura del teclado, el programa comienza poniendo en estado alto (uno lógico) el pin de salida que gobierna la primera fila del teclado, como muestra la Fig. 8 (a). Luego de esto, el programa realiza la lectura de los pines asociados a las columnas y si ningún

pulsador es presionado, como en la Fig. 8 (a), las entradas del microcontrolador reciben un estado bajo (cero lógico). Al presionar alguna de los pulsadores de la primera fila, por ejemplo, KEY12 en la Fig. 8 (b), el estado alto en la primera fila ocasiona una corriente a través de la resistencia R2, llegando un estado alto al segundo pin de entrada del microcontrolador, indicando al programa que el pulsador mencionado está presionado. Si el programa pone en alto la cuarta fila y es presionado el pulsador KEY42 como lo muestra la Fig. 8 (c), el programa vuelve a leer un estado alto en el segundo pin de entrada del microcontrolador.

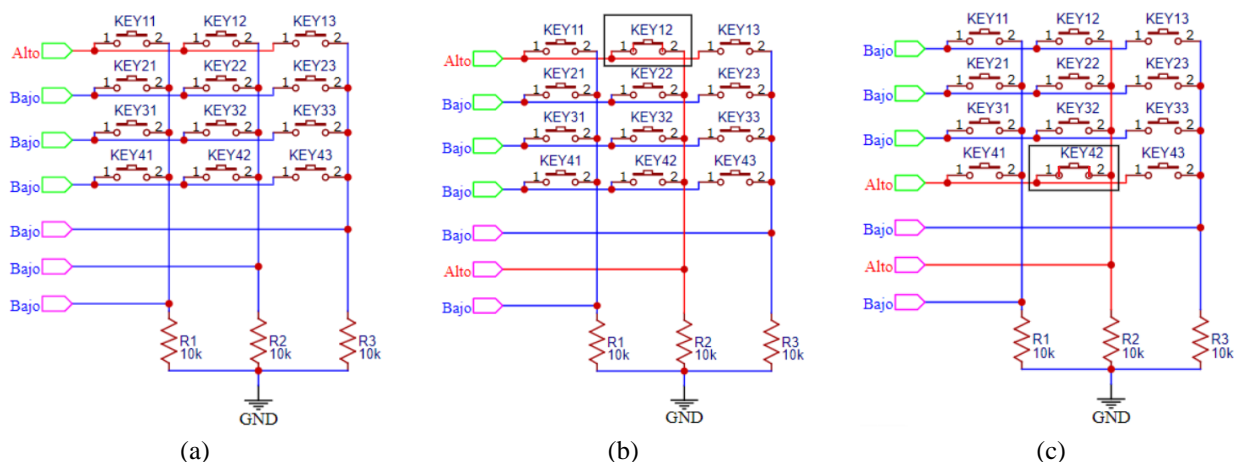


Fig. 8. Lectura de pulsadores del teclado: (a) Primera fila sin pulsador presionado; (b) Primera fila con KEY12 presionado; (c) Cuarta fila con KEY42 presionado.

Comparando los dos casos de lectura de pulsadores presentados, puede concluirse que a través de los pines de entrada del microcontrolador no puede obtenerse el estado de cada pulsador directamente, razón por la cual no pueden ponerse en estado alto las cuatro filas simultáneamente. De esta forma, el modo de trabajo del microcontrolador con el teclado matricial es la siguiente:

- 1) El programa activa la primera fila del teclado.
- 2) Luego el mismo realiza la lectura de los pines de entrada para saber el estado de cada pulsador de la fila y se realiza la acción correspondiente en caso de que alguno se encuentre presionado.
- 3) El programa realiza mismo procedimiento para la fila siguiente.
- 4) Al terminar de leer la cuarta fila, se vuelve a la primera.

Esta sucesión de lecturas de cada fila es realizada continuamente y a gran velocidad por el programa, para dar al usuario la sensación de tener activas todos los pulsadores al mismo tiempo. La Fig. 9 muestra el diagrama de flujo del programa para realizar la lectura del teclado.

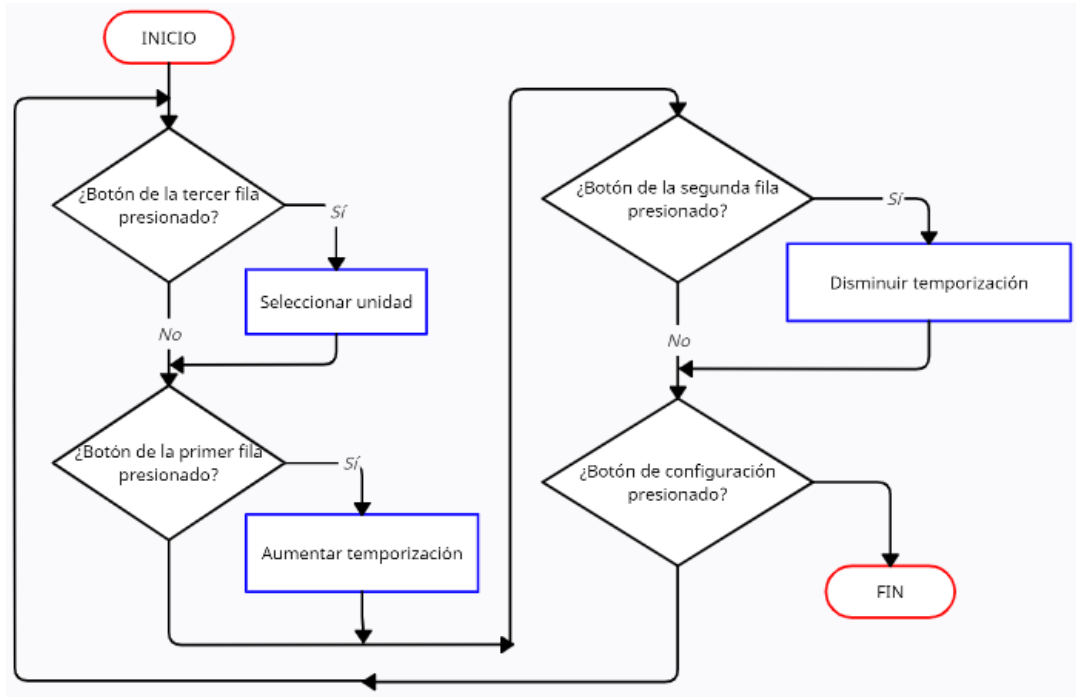


Fig. 9. Diagrama de flujos para la lectura del teclado en el modo Configuración.

Cabe mencionar que para ingresar al modo Configuración, en el teclado del TP debe presionarse el pulsador Configuración mostrado en la Fig. 6. Esto puede observarse en la parte final del diagrama de flujos anterior.

2.3. Interfaz de potencia

Esta sección explica el modo de conexión de los dos relés del sistema, controlados por el TP. Estos relés no son conectados directamente a los pines de salida del microcontrolador, se los conectan a través de transistores bipolares NPN, los cuales son usados como interfaz de corriente [7]. La Fig. 10 muestra la conexión mencionada.

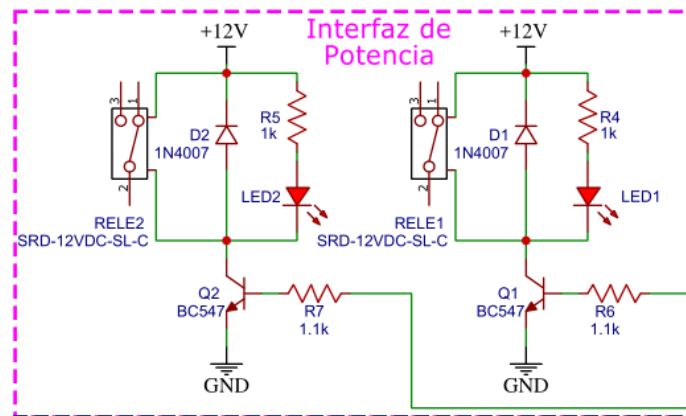


Fig. 10. Circuito de la interfaz de potencia asociada a los relés.

En la Fig. 10 se observan los resistores R6 y R7 encargados de limitar la corriente que ingresa a la base de los transistores Q1 y Q2 respectivamente, los cuales operan en corte y saturación. Los transistores son protegidos a través de diodos de rueda libre que están en antiparalelo con la bobina de cada uno de los relés. También en paralelo con cada bobina hay un LED para indicar si el relé esta encendido o apagado.

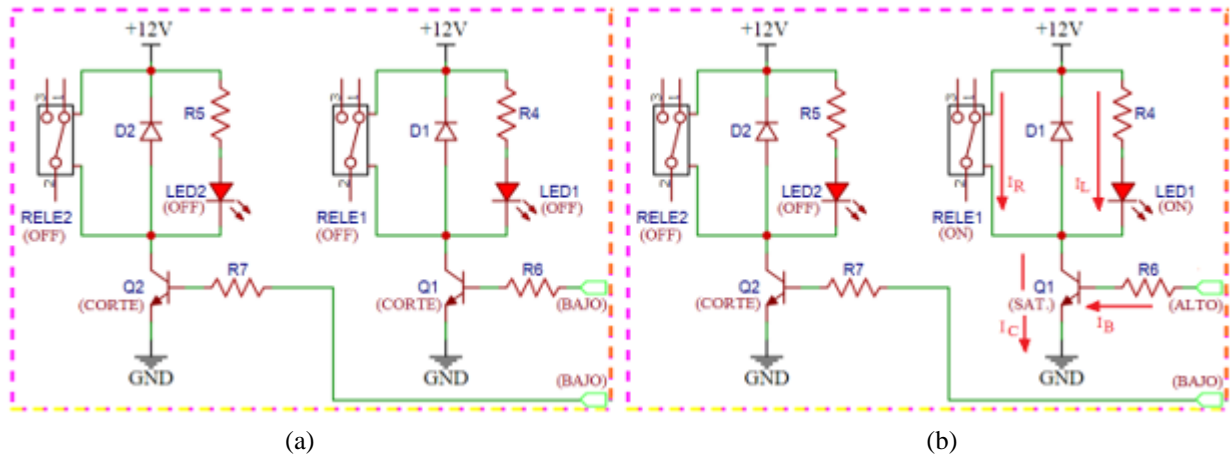


Fig. 11. Accionamiento de los relés: (a) Relés desconectados; (b) Relé 1 encendido.

Cuando el Temporizador Programable debe encender determinado relé, el programa del microcontrolador pone un estado alto en la resistencia de base del transistor correspondiente y de esta forma inyecta en este dispositivo una corriente que ocasiona su saturación. Atendiendo a esto, la Fig. 11 (a) presenta la situación en la que ningún relé se encuentra encendido. En este caso los dos pines de salida del ATmega 2560 entregan un estado bajo ($\approx 0V$). Por lo tanto, los transistores se encuentran en estado de corte y no permiten la circulación de corriente a través de las bobinas de los relés y sus correspondientes LEDs. En el caso de la Fig. 11 (b) se energiza al Relé 1; para esto, el microcontrolador entrega un estado alto ($\approx 5V$) a través del pin de salida correspondiente, produciendo una circulación de la corriente de base I_B a través del resistor R6 y dejando al transistor Q1 en estado de saturación. Esto permite que circule corriente desde la fuente de +12V a través de la bobina del relé y del LED indicador, siendo estas corrientes I_R e I_L respectivamente. La suma de estas dos corrientes, I_C , circula a través del colector del transistor Q1.

A partir de la operación mencionada, cuando el programa da inicio a la temporización de alguno de los relés, la salida correspondiente del microcontrolador entrega un estado alto al transistor correspondiente. Cuando termina el tiempo programado, el pin de salida mencionado pasa a un estado bajo y apaga el relé con su respectivo LED.

La Tabla 1 muestra las especificaciones técnicas de los componentes utilizados en el circuito de la Fig. 10. Las misma son utilizadas para el dimensionamiento de las resistencias indicadas en dicho circuito, que a continuación se describe.

Tabla 1. Especificaciones técnicas [1], [8], [9].

ATmega 2560		
Tensión de salida en estado alto	V_{OHmin}	4,2 V
Corriente de salida en estado alto	$I_{Omáx}$	20 mA
Transistor BC547 (Q1 y Q2)		
Ganancia de corriente	$h_{fe\ min}$	125
Tensión colector-emisor de sat.	$V_{CE(sat)máx}$	0,25 V
Tensión base-emisor de encendido	$V_{BE(on)máx}$	0,7 V
Corriente de colector máx.	$I_{Cmáx}$	100 mA
SRD-12VDC-SL-C (Relé 1 y Relé 2)		
Tensión nominal	V_{RN}	12 V
Corriente nominal	I_{RN}	30 mA
LED Rojo (D1 y D2)		
Caída de tensión en directa	V_{LED}	2,2 V
Corriente de polarización en directa	I_{LED}	10 mA

A través de la Ec. (2) se calculan las resistencias R4 y R5 que limitan la corriente a través de cada LED de la interfaz de potencia indicada en la Fig. 10.

$$R = \frac{(V_{CC} - V_{LED} - V_{CE(sat)máx})}{I_{LED}} ; P = F_s \cdot I_{LED}^2 \cdot R \quad (2)$$

Siendo $V_{CC} = 12\text{ V}$ y con los valores de la Tabla 1, se obtiene $R = 955\Omega$. A partir de este valor, para R4 y R5 se adopta el valor comercial de 1 k Ω . Utilizándose un factor de seguridad $F_s = 2,5$ para sobredimensionar la resistencia en caso de un calentamiento excesivo en el circuito, se obtiene $P = 0,22\text{ W}$. A partir de esto, para las resistencias R4 y R5 se adopta un valor comercial de 1 k Ω / 1/4W.

Los valores de las resistencias de base R6 y R7 mostradas en la Fig. 10 son calculados a partir de la corriente que debe manejar el colector de cada transistor dada por la siguiente suma:

$$I_C = I_{RN} + I_{LED} \quad (3)$$

El valor de la corriente de colector es $I_C = 39,5\text{ mA}$, siendo inferior al valor $I_{Cmáx}$ indicado en la Tabla 1 para el transistor utilizado. Con la Ec. (5) y a partir de I_C puede obtenerse la corriente de base I_B necesaria para saturar a cada transistor.

$$I_B = 10 \frac{I_C}{h_{fe\ min}} \quad (4)$$

Considerando $h_{fe\ min} = 125$ y la corriente de colector I_C que debe manejar cada transistor, la corriente de base para la saturación de los mismos resulta en $I_B = 3,2\text{ mA}$. Como puede apreciarse en la Tabla 1, este valor de corriente es inferior a la corriente de salida máxima $I_{Omáx}$ que puede entregar cada pin del ATmega 2560, indicando que resulta adecuado el uso de los transistores Q1 y Q2.

Con la corriente I_B la resistencia de base de cada transistor puede dimensionarse a partir de la siguiente ecuación:

$$R = \frac{(V_{OHmín} - V_{BE(on)máx})}{I_B} ; P = F_S \cdot I_B^2 \cdot R \quad (5)$$

Considerando las especificaciones de la Tabla 1, la resistencia de base resulta $R = 1,09 \text{ k}\Omega$ adoptándose para R6 y R7 el valor comercial de $1,1\text{k}\Omega$. Con un factor de seguridad $F_S = 2,5$, la potencia para estas resistencias resulta $P = 0,028 \text{ W}$. A partir de esto, para R6 y R7 se adopta un valor comercial de $1,1 \text{ k}\Omega / 1/4\text{W}$.

2.4.Display LCD

La visualización continua del funcionamiento del programa del Temporizador Programable se hace a través de un display de cristal líquido (LCD), capaz de la representación de caracteres alfanuméricos. El modelo utilizado es Display LCD 1602A con Blacklight azul. Este dispositivo presenta dos filas y 16 columnas, pudiendo representar en total 32 caracteres a la vez. La Fig. 12 muestra cómo se interconecta este display al microcontrolador. Como puede apreciarse, el microcontrolador es conectado al mismo a través de seis pines, de PC2 a PC7, configurados como salida. El preset indicado en la Fig. 12 permite al usuario el ajuste del contraste del display en forma manual [10].

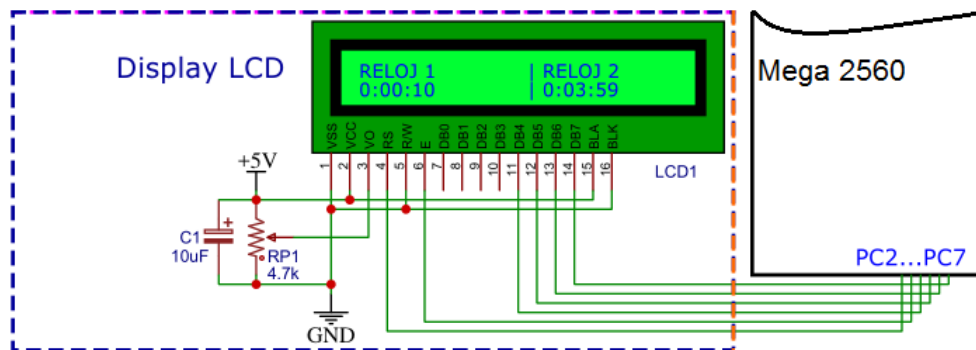


Fig. 12. Circuito de conexión entre el display LCD y el ATmega 2560.

En el Temporizador Programable, el display muestra el tiempo faltante para la finalización de la temporización de cada relé. En caso de que ambos relés permanezcan encendidos, el display indica el tiempo restante de los dos. La Fig. 12 muestra esta visualización, del lado izquierdo del display aparecen los datos del relé 1, mientras que del lado derecho los datos del relé 2.

Durante el proceso de programación de las temporizaciones, el display indica los valores correspondientes para que el usuario pueda realizar este procedimiento.

3. Resultados

Dado que todo el desarrollo de este trabajo se ha realizado fuera de los laboratorios de la facultad, debido a las restricciones a la presencialidad que hubo durante el cursado del año 2020, el circuito del sistema fue desarrollado a modo de prototipo experimental utilizándose los materiales disponibles.

Inicialmente se desarrolla y comprueba el correcto funcionamiento del programa mediante el software de simulación Proteus. Luego se arma el teclado con los doce pulsadores, ordenados en cuatro filas y tres columnas, y una caja de cartón. Del lado interior de la caja se sueldan los pulsadores mediante cables y estaño. Esto puede apreciarse en la siguiente Fig. 13. Gracias al teclado armado con los pulsadores, se obtuvo un funcionamiento estable del teclado. En el caso de haber montado los pulsadores en el protoboard, probablemente se hubieran tenido problemas para configurar y controlar el TP debido a falsos contactos.



Fig. 13. Teclado construido a partir de pulsadores.

Seguidamente el circuito propuesto en la Fig. 1 es montado en el protoboard, resultando lo mostrado en la Fig. 14. Para alimentar este circuito es utilizada las tensiones de +5V y +12V de una fuente de PC de escritorio.

La configuración de las dos temporizaciones, así como la puesta en marcha y parada de los relés, funcionan tal como se esperaba de acuerdo al código del programa. Como se estableció anteriormente, a través del display pueden observarse los diferentes estados de ambos relés. Las temporizaciones pueden visualizarse al mismo tiempo y configurar los tiempos de encendido uno a la vez.

Para verificar el correcto funcionamiento de los temporizadores, fueron configurados diversos tiempos de funcionamiento, con un máximo de 45 minutos. Una vez puestos en marcha, se controló el tiempo mediante el temporizador de un smartphone con SO Android. Ambos relés entran funcionamiento al iniciar la cuenta correspondiente, y se apagan una vez transcurrido el tiempo establecido para cada uno de ellos.

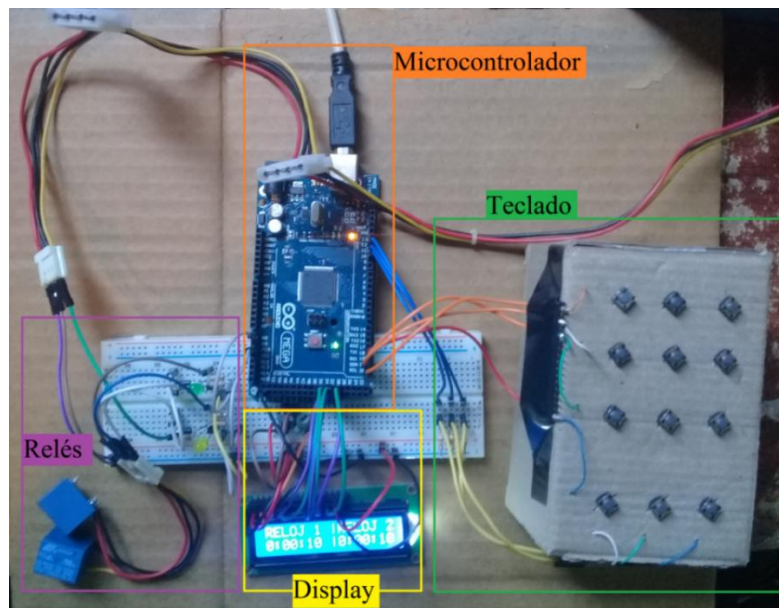


Fig. 14. Prototipo experimental del Temporizador de Intervalos Programables.

La prueba del circuito se realizó mediante videoconferencia en presencia de los docentes la asignatura Técnicas Digitales 2. Durante la misma se comprobó el correcto funcionamiento del Temporizador Programable desarrollado. El sistema cumple con los requerimientos detallados anteriormente, siendo la respuesta y el comportamiento del circuito el establecido por el código C del programa realizado.

4. Conclusiones

A través de los ensayos realizados se comprobó que el sistema desarrollado permite realizar la temporización programada en el accionamiento de dos relés, siendo esta temporización programable en forma independiente para cada uno ellos. Esto permite decir que el Temporizador de Intervalos Programables obtenido cumple con las especificaciones propuestas en el Trabajo Integrador de la asignatura Técnicas Digitales 2.

Por otra parte, el circuito desarrollado puede resultar de gran utilidad en sistemas o circuitos eléctricos que funcionen durante un periodo de tiempo limitado y conocido. Puede usarse como complemento de cargadores de notebooks o celulares, para usar energía eléctrica solamente mientras el dispositivo está en proceso de carga. Otro ejemplo de uso puede ser la temporización en la operación de bombas de agua para el llenado de tanques de agua o piletas. El uso del TP desarrollado también puede tenerse en cuenta con en termotanques o demás sistemas de calentamiento.

Si bien el temporizador diseñado cumple con las funciones solicitadas, hay varias mejoras que pueden implementarse a futuro. Para realizar la configuración de la temporización de cada relé, es necesario que se encuentren apagados. Si alguno está encendido y desea configurarse un nuevo tiempo de funcionamiento, debe detenerse la cuenta para realizar una nueva configuración. Para salvar esto, puede agregarse una función que permita pausar la cuenta de tiempo a través del teclado sin desenergizar el relé, pudiéndose configurar el nuevo tiempo con el teclado sin necesidad de detener primero el relé correspondiente. También puede modificarse la toma de datos del teclado,

permitiendo aumentar o disminuir el tiempo en configuración mediante la presión continua de los pulsadores, sin necesidad de presionarlas repetidamente, y así prolongar la vida útil de los mismos. Un agregado de mucha utilidad sería incorporar una temporización extra a cada relé, para también contar con un encendido programado de cada uno.

Este trabajo sirvió como una experiencia en la solución de un problema tecnológico, partiéndose de una necesidad con ciertas especificaciones técnicas. Lo dicho fue muy importante, ya que al tener que desempeñarnos como ingenieros en el futuro, deberemos dar respuesta a diversos problemas, donde se reciben solamente algunas especificaciones técnicas y de operación. Igual que con este problema en particular, la solución y la manera de llevarla a cabo son elegidas a partir de los conocimientos y la experiencia de la persona encargada.

Para el diseño y puesta en marcha del Temporizador de Intervalos Programables, fue necesario programar el microcontrolador, aplicando los conocimientos obtenidos en Técnicas digitales 2. Para el desarrollo de la Interfaz de Potencia se utilizaron conocimientos de las asignaturas Dispositivos Electrónicos y Técnicas Digitales 1. Todo esto pone de manifiesto la importancia de este tipo de actividades, las cuales ponen en juego la integración de los saberes adquiridos durante la carrera.

Referencias

- [1] CompraRacional La web de los compradores con criterio. “Los 7 mejores temporizadores: ahorra energía y libérate de preocupaciones” (2019, Agosto 27). Disponible: <https://www.compraracional.com/los-mejores-enchufes-temporizadores/>
- [2] Arduino en español. “Arduino Mega 2560” (s.f.). Disponible: <http://manueldelgadocrespo.blogspot.com/p/arduino-mega-2560.html>
- [3] Mgter. Ing. Fernández Guillermo A. “Microcontroladores – ATmega2560. Características, Instrucciones y Modos de Direccionamiento” presentación de ET446-Técnicas Digitales 2, Dpto. de Ingeniería Electrónica, Fac. de Ingeniería, UNaM.
- [4] Atmel Corporation. “Atmel Studio USER GUIDE”, guía de usuario Atmel-42167B-Atmel-Studio_User Guide-09/2016, (2016). Disponible: http://ww1.microchip.com/downloads/en/devicedoc/atmel-42167-atmel-studio_user%20guide.pdf
- [1] Atmel Corporation. “Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V”, datasheet 2549Q-AVR-02/2014, (2014). Disponible: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf
- [6] Geek Factory. “Teclado Matricial con PIC” (2013, Julio 26). Disponible: <https://www.geekfactory.mx/tutoriales/tutoriales-pic/teclado-matricial-con-pic/>
- [7] Mgter. Ing. Fernández Guillermo A. “Familias Lógicas de Circuitos Integrados Digitales” presentación de ET442-Técnicas Digitales 1, Dpto. de Ingeniería Electrónica, Fac. de Ingeniería, UNaM.
- [8] Motorola Inc. “Amplifier Transistors NPN Silicon”, datasheet BC546/D, (1996). Disponible: <https://pdf1.alldatasheet.es/datasheet-pdf/view/11551/ONSEMI/BC547.html>
- [9] Ningbo Songle Relay Co. “SRD Series SUBMINATURE HIGH POWER RELAY 7A/10A/15A”, datasheet (s.f.). Disponible: <http://www.songlerelay.com/Public/Uploads/20161104/581c81ac16e36.pdf>
- [10] “Actividad Experimental N°3: Uso del Conversor Analógico-Digital en Microcontroladores ATmega2560” presentación de ET446-Técnicas Digitales 2, Dpto. de Ingeniería Electrónica, Fac. de Ingeniería, UNaM.