



Sistema para mando de relés a distancia usando control remoto de TV

Guillermo A. Centurión Barrero ^{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.*

guilleaxelb2@gmail.com, gross@fio.unam.edu.ar, fernandez@fio.unam.edu.ar

Resumen

Este artículo presenta los resultados del Trabajo Integrador realizado en la asignatura Técnicas Digitales 2, de cuarto año de la carrera Ingeniería Electrónica. En el mismo se desarrolla un sistema embebido basado en el microcontrolador ATmega2560, incluido en la placa de desarrollo Arduino Mega. El circuito construido permite comandar a distancia dos relés, utilizando cualquier control remoto de TV con salida infrarroja. Los relés pueden operar en modo on/off o temporizado, siendo esto configurable por el usuario. El circuito propuesto posee un display LCD y dos potenciómetros para configurar el modo de operación de los relés. Inicialmente en este trabajo es estudiada la operación de distintos protocolos de comunicación de los controles remotos. A partir de esto se diseña el sistema embebido, comenzando por establecer el funcionamiento deseado del circuito, luego desarrollando el programa para el microcontrolador y calculando los componentes necesarios para el circuito. Seguidamente el mismo es montado en un protoboard y se comprueba su funcionamiento correcto. A partir de los resultados obtenidos, puede concluirse que el sistema desarrollado es sencillo y adaptable para casi cualquier control remoto, pudiéndose controlar con bastante fiabilidad artefactos del hogar o en maquinarias de uso industrial algunos casos, lo cual cumple los objetivos del Trabajo Integrador.

Palabras Clave – Arduino; microcontrolador; control; mando; relé; infrarrojo; microcontrolador; programación.

1. Introducción

La actividad integradora de la cual resulta este trabajo, no sólo tiene por objetivo articular los saberes adquiridos durante el cursado de la asignatura Técnicas Digitales 2, sino que también diseñar y construir un sistema embebido que sea capaz de encender y apagar de manera remota una variedad de dispositivos, tanto domésticos como industriales tales como ventiladores, luces, heladeras, etc.(domésticos) o cintas transportadoras, motores, luces, etc.(industriales).

El circuito de este trabajo requiere de un control remoto con comunicación de datos mediante señales de luz infrarroja (IR). La diferencia entre estos controladores y los que usan señales de radiofrecuencia (RF), es que las señales de infrarrojo tienen un alcance de apenas unos pocos metros en comparación con los que operan con las señales de radiofrecuencia. Además, los controles remotos con IR emiten señales que son “direccionales”, es decir, es necesario que el emisor de la señal

* Guillermo A. Centurión Barrero.

infrarroja apunte en la dirección del receptor. Si la señal es muy débil (por ejemplo, por batería baja en el mando), la comunicación de datos sin errores es más sensible a la alineación del emisor con respecto al receptor [1].

A diferencia de la señal infrarroja, las de radiofrecuencia pueden atravesar obstáculos (como ser paredes y puertas) y llegan más lejos. Los controles remotos que operan con señales de RF son más utilizados cuándo es necesario un rango de acción superior, por ejemplo, en mandos a distancia de puertas automáticas, sistemas de alarma o juguetes de radio-control. Este mayor alcance es debido a que las señales de RF tienen una mayor longitud de onda que las señales de IR [1].

Frente a las desventajas mencionadas, se destaca que los controles remotos con IR consumen mucha menos energía que los de RF. Además, son más económicos y usualmente son los que se utilizan en los dispositivos domésticos por lo que es fácil acceder a uno. Por otra parte, las señales de RF pueden causar interferencias y afectar a otros dispositivos [2].

Atendiendo a las cuestiones indicadas y como es deseable utilizar un control remoto económico, fácil de conseguir y de utilizar, en el desarrollo de este trabajo se ocupa uno que opera con señales IR. El sistema embebido desarrollado se basa en el microcontrolador ATmega2560 [3], disponible en la placa de desarrollo Arduino Mega. Mediante dos salidas digitales, el microcontrolador mencionado gobierna el estado de dos relés. Para efectuar el accionamiento de los mismos, el programa desarrollado para el microcontrolador recibe los comandos enviados por el usuario a través de un control remoto con salida infrarroja. Este dispositivo puede operar con alguno de los protocolos de comunicación infrarroja (IR) comunes (NEC [4], JVC [5], NRC17 [6], RC-5 [7], SIRC [8] u otros), ya que, en el sistema desarrollado esto puede ser programado por el usuario.

Los controles remotos que poseen televisores, aires acondicionados, equipos de audio y otros, envían mensajes en forma de una sucesión de impulsos de luz infrarroja codificada en binario. En la mayoría de los protocolos de comunicación IR que son usados en estos artefactos, para poder diferenciar si el control remoto envió un 1 (alto) o un 0 (bajo), el receptor mide el tiempo de duración de estos estados. A partir de esto, el circuito propuesto en este trabajo posee un modo de programación donde el usuario puede registrar (memorizar) el código de algunas teclas del control remoto que desea utilizar para comandar los relés. Para estos últimos, el usuario puede configurar el modo de operación que poseen ante los comandos enviados por el control remoto utilizado. Estos modos de operación son:

- **On/Off:** En este modo, cuando el usuario presiona la tecla correspondiente del control remoto (previamente programada en el sistema), se activará el relé en cuestión. Luego, si el usuario presiona nuevamente la misma tecla, el relé correspondiente se apagará.
- **Temporizado:** Para esta forma de operación, el relé se enciende por un tiempo y luego se apaga. Este tiempo puede ser programable a través de una tensión variable proporcionada por un potenciómetro. Cada relé tiene su potenciómetro.

El sistema propuesto emplea el receptor infrarrojo TL1838 [9], para establecer la comunicación entre el control remoto y el microcontrolador. Este receptor permite extraer de la señal IR los valores digitales correspondientes a los comandos generados al presionar las teclas del control remoto utilizado.

Por otro lado, el sistema embebido cuenta con un display LCD, que actúa como interfaz de usuario para configurar los modos de operación de los relés y también programar las teclas del control remoto que serán usadas para comandar a los mismos.

Para la realización del programa que ejecuta el microcontrolador ATmega2560, se utiliza el entorno de desarrollo Atmel Studio [10], que permite programar a este dispositivo utilizando el lenguaje C.

En base a lo explicado, seguidamente se presenta el sistema desarrollado y la fundamentación de su funcionamiento, tanto desde el punto de vista del software como del hardware. Luego se exponen los cálculos y consideraciones correspondientes al diseño de las etapas del sistema, mostrándose finalmente los resultados obtenidos y las conclusiones correspondientes.

2. Diseño del sistema embebido

2.1 Firmware del microcontrolador

Los controles remotos usados habitualmente en artefactos del hogar, envían señales luminosas que están en el espectro infrarrojo, invisible al ojo humano. Las señales mencionadas están constituidas por una serie de pulsos, por lo general de una frecuencia de 38 kHz, ya que a esta frecuencia no se encuentran muchas perturbaciones del ambiente que podrían afectar a la transmisión de los datos [2]. A esta señal se la denomina portadora y está modulada por los datos que son transmitidos. Esto lo muestra la Fig. 1.

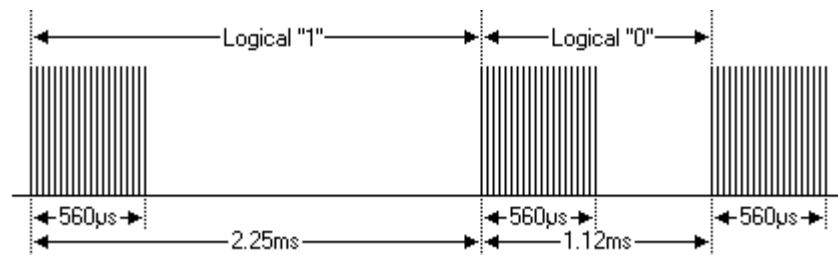


Fig. 1. Modulación utilizada para enviar datos con el protocolo NEC [4].

Como se muestra en la Fig. 1. La información es enviada cambiando la duración de los tiempos en bajo mientras mantiene los tiempos en alto constantes, esto aplica al protocolo utilizado en esta figura (protocolo NEC [4]). Pero también hay otros protocolos que envían la información cambiando la duración de los tiempos en alto (protocolo Philips RC-5 [7]) o modificando tanto los tiempos en alto como los bajos (protocolo Sony SIRC [8]). Estas cuestiones se han considerado en el desarrollo del programa para el microcontrolador, con el fin de que el sistema propuesto pueda operar con una variedad de protocolos usados en los controles remotos más populares.

El receptor infrarrojo TL 1838 utilizado en este trabajo, posee un fotodetector que es sensible a las señales infrarrojas con una portadora de frecuencia igual a la indicada. Este dispositivo también incorpora un circuito capaz de demodular a la portadora, entregando a su salida pulsos de distinta

duración que constituyen los estados lógicos de los datos transmitidos. De esta forma, una señal como la indicada en la Fig. 1, llega hasta el receptor y el mismo elimina el tren de pulsos de 38 kHz, dejando únicamente los estados alto y bajo que modulan a la portadora. La duración de estos estados proporciona la información de importancia para el circuito conectado al receptor (es decir, los 0s y 1s de los datos transmitidos), en este caso el microcontrolador encargado de interpretar la información transmitida al presionar cada tecla del control remoto que es utilizada en el comando de los relés.

La duración de los pulsos para indicar cada estado lógico (0 o 1) de la información transmitida, puede variar según el protocolo que utilice el control remoto, pero en su mayoría están en torno a unos pocos milisegundos [2]. Por este motivo, el *firmware* (programa) desarrollado para que el microcontrolador interprete la duración de los estados lógicos presentes en la señal enviada por el control remoto, utiliza el módulo temporizador denominado Timer 0 que posee el ATmega2560. Este recurso permite medir la duración de los estados lógicos mostrados en la Fig. 1.

Para realizar la medición de la duración de tiempo indicadas, el Timer 0 es configurado para operar en modo normal. De esta forma, su contador $TCNT0$ es incrementado a partir de la señal de reloj con que opera el microcontrolador y de esta forma, la Ec. (1) representa el intervalo de tiempo en el que $TCNT0$ cambia en una cuenta.

$$T = \frac{PRE}{F_{CLK}} \quad (1)$$

donde F_{CLK} es la frecuencia de la señal de reloj que posee el microcontrolador y PRE corresponde al valor del prescaler usado en el Timer 0. Siendo $F_{CLK} = 16$ MHz y $PRE = 1024$, el contador $TCNT0$ del temporizador, es incrementado cada $T = 64$ μ s. Considerando esto, la Ec. (2) permite establecer la relación que hay entre las cuentas obtenidas en $TCNT0$ y el “Tiempo” de duración de los estados lógicos 0 y 1 de la señal recibida por el microcontrolador, desde el receptor infrarrojo.

$$TCNT0 = \frac{Tiempo}{T} \quad (2)$$

En el caso de la señal indicada en la Fig. 1, el contador del Timer 0 obtiene un valor $TCNT0 = 9$ cuando el microcontrolador recibe un estado lógico “0” y de 27 cuando recibe un estado lógico “1”. De esta manera, el programa que ejecuta el microcontrolador determina los “0s” y “1s” que posee la señal transmitida por el control remoto, con sólo analizar el valor que toma el contador $TCNT0$ del temporizador Timer 0.

La Fig. 2 muestra la salida del receptor TL 1838 cuando recibe una trama transmitida con el protocolo NEC. Como se ha mencionado antes, el receptor elimina los pulsos de la portadora de 38 kHz y demodula a la misma, siendo la señal indicada en la figura la que se recibe en la entrada del microcontrolador, cuando el usuario presiona una tecla de un control remoto que opera con este protocolo de comunicación.

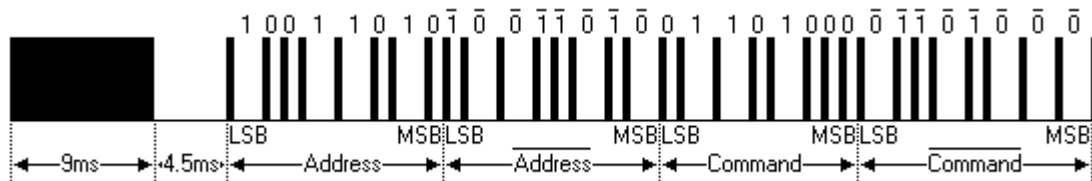


Fig. 2. Estructura de una trama para el protocolo NEC [4].

La trama de la Fig. 2 comienza con un estado en alto (9 ms) seguido por un estado bajo (4,5 ms), ambos de larga duración en comparación a la parte correspondiente a los datos de la trama. Esto permite que el programa del microcontrolador pueda detectar cuando comienza cada trama, acción que se realiza comprobando periódicamente el estado lógico que toma la entrada del microcontrolador conectada a la salida del TL 1838. A los estados de inicio de trama le siguen la dirección y el comando. Ambos transmitidos dos veces, siendo la segunda vez con todos los bits invertidos para que el receptor pueda comprobar errores en el mensaje recibido [2].

El protocolo NEC es ampliamente usado en los controles remotos universales, pero debido a que el sistema desarrollado debe ser compatible con controles que poseen otros protocolos, en el *firmware* desarrollado se ignoran los valores correspondientes a las direcciones y comandos ya que estos constituyen información propia del protocolo de comunicación utilizado. El *firmware* sólo interpreta la duración de los estados lógicos 0 y 1 que posee la envolvente de la señal mostrada en la Fig. 1 y a partir de estos obtiene el código binario correspondiente a la tecla pulsada en el control remoto. Como la duración de cada estado lógico de la envolvente mencionada puede variar entorno a algunos microsegundos, en la medición de tiempo realizada con el Timer 0, se incorpora un margen de tolerancia.

Según lo analizado en distintos protocolos, la cantidad de bits de datos enviada en una trama no supera 32 bits. Por lo cual, en la parte del *firmware* dedicada a la programación de las teclas del control remoto, los tiempos de duración de los estados lógicos 0 y 1 presentados en la envolvente de los datos de la trama, son almacenados en dos vectores de 32 elementos cada uno. A esto se los denominados “vectores de mediciones”. Cada uno de los elementos de estos vectores es una variable de 16 bits cargada con el contenido del contador *TCNT0*, luego de cada una de las 64 mediciones realizadas para obtener los tiempos de duración de los estados lógicos que posee la señal demodulada. El proceso de almacenamiento indicado, son ignorados los tiempos en alto y bajo que dan inicio a la trama.

Una vez medidos y almacenados los tiempos de duración que poseen los estados lógicos de la señal demodulada, el programa del microcontrolador determina la correspondencia entre estos tiempos y los 1s y 0s que posee como datos la señal indicada. Para realizar esta correspondencia, el programa toma como referencia el primer elemento del vector de mediciones (tiempo de duración del primer bit de datos de la trama) y compara su valor con cada uno de los valores que posee el resto de los elementos del vector. Siendo el valor del primer elemento $T1$ y el valor de cualquiera de los elementos restantes Ti , si en alguna de las sucesivas comparaciones resulta $Ti > (T1+2)$, entonces al primer elemento del vector de mediciones le corresponde un “0”. Ahora si en alguna de las sucesivas

comparaciones resulta $T_i < (T1-2)$, al primer elemento le corresponde un “1”. De esta forma queda determinada la correspondencia que hay entre la duración “larga” y cada “0” de la trama, como así también la correspondencia entre la duración “corta” y cada “1” de la trama. Esto se hace tanto para los tiempos en alto como para los tiempos en bajo ya que como se mencionó antes, dependiendo del protocolo los datos se envían modulando la información en los tiempos en alto o en los tiempos en bajo (o en ambos). Cabe aclarar que la afectación de dos unidades al valor de T1 en la comparación, es para proporcionar una tolerancia y así contemplar la variabilidad existente en los tiempos medidos. Este procedimiento permite la conversión de los tiempos medidos en la trama a un código binario representativo de la tecla que se desea programar para el accionamiento de alguno de los relés. El procedimiento es realizado en el programa debido a que el microcontrolador no almacena los patrones de duración correspondientes a los estados lógicos que posee cada protocolo de comunicación existente, lo cual no sería práctico y tampoco contribuiría a que el sistema propuesto pueda adaptarse fácilmente a diversos protocolos utilizados en los controles remotos.

El vector de mediciones utilizado en la conversión explicada posee 32 elementos debido a que se considera que la trama del protocolo correspondiente a un control remoto estándar, tiene como máximo 32 bits de datos. En caso de que la trama posea menor cantidad de bits de datos, el vector mencionado sigue teniendo la misma longitud. Pero aquellos elementos que corresponden a bits de datos inexistentes en la trama, estarán cargados con valor cero dado que en las mediciones de duración de tiempo realizadas con el Timer 0, su contador *TCNT0* se desbordará cargándose con “0” aquellos valores mayores a 255. De esta forma, en el código binario obtenido de la conversión mencionada, a cada elemento “0” del vector de mediciones el programa desarrollado le asigna un 0 en dicho código.

Todo lo explicado anteriormente constituye parte del *firmware* dedicado a programar la tecla del control remoto usada para el accionamiento del relé en cuestión. Dado que esto es una parte crítica, para realizarlo el usuario debe pulsar dos veces la misma tecla. De esta forma, el programa desarrollado obtiene dos veces el código binario de una tecla, comprueba si los mismos son iguales y en caso de serlo, el código correspondiente a la tecla es almacenado en la memoria de dato no volátil que posee el microcontrolador. Con este procedimiento de doble lectura/conversión, el sistema comprueba que no existan errores en la transmisión de los datos, pudiendo así programarse correctamente la tecla del control remoto. Durante la programación de la tecla, el display LCD que posee el sistema desarrollado, indica al usuario si la doble lectura/conversión de la tecla ha sido correcta o no. En caso de que los dos códigos binarios obtenidos para una misma tecla sean iguales y fue almacenado el mismo en la memoria no volátil, el display muestra el mensaje “Guardado completo”. Pero en caso de que no sean iguales los códigos, el display muestra un mensaje de “Error al guardar” y el usuario debe volver a realizar todo el procedimiento de programación de la tecla.

Para programar una nueva tecla y también seleccionar el modo de operación de los relés, el sistema propuesto posee una interfaz de usuario conformada por tres 3 pulsadores (denominados 1, 2 y 3) y un display LCD. La Fig. 3 muestra la operación de esta interfaz de usuario. En el diagrama de dicha figura aparecen los mensajes que van mostrándose en el display a medida que el usuario selecciona las diferentes opciones con los pulsadores correspondientes. Al seleccionarse la opción 1, el sistema pregunta qué relé es el que desea controlarse; una vez seleccionado el relé, el sistema solicita presionar dos veces la tecla del control remoto a utilizada para comandar el relé seleccionado.

Seleccionándose la opción 2, puede accederse a configurar el modo de operación del relé correspondiente. Con la selección de la opción 3 puede borrarse el código de una de las teclas programadas anteriormente. Cabe mencionar, que sólo se puede programar el código de dos teclas del control remoto, una para cada relé que posee el sistema propuesto.

Los relés que posee el sistema desarrollado pueden operar en dos modos y la configuración de esto puede hacerse a través de la opción 2 indicada en el recuadro superior de la Fig. 3. Los modos posibles son “on/off” y “temporizado”. En el modo “on/off”, cada vez que el usuario pulsa la tecla programada para el relé correspondiente, el mismo cambia de estado encendiéndose o apagándose. Para configurar este modo de operación, una vez ingresado a la opción 2 mencionada, debe seleccionarse el relé correspondiente y luego el modo de operación “on/off”. En el modo “temporizado”, cada vez que el usuario presiona la tecla del relé correspondiente, el mismo se enciende durante un cierto tiempo. Para configurar este modo de operación, además de seleccionar este modo como en caso anterior, también debe programarse el tiempo que permanecerá encendido el relé en cuestión. Esto se hace ajustando el voltaje que entregan los potenciómetros del circuito, los cuales están conectados a los canales analógicos ADC3 y ADC4 que posee el conversor analógico-digital (ADC) del microcontrolador. El conversor opera en modo *free running* y entrega un valor digital que el programa convierte en un número entre 0 y 10, correspondiendo esto al rango de temporización en segundos que puede configurarse en el sistema para este modo de operación de los relés.

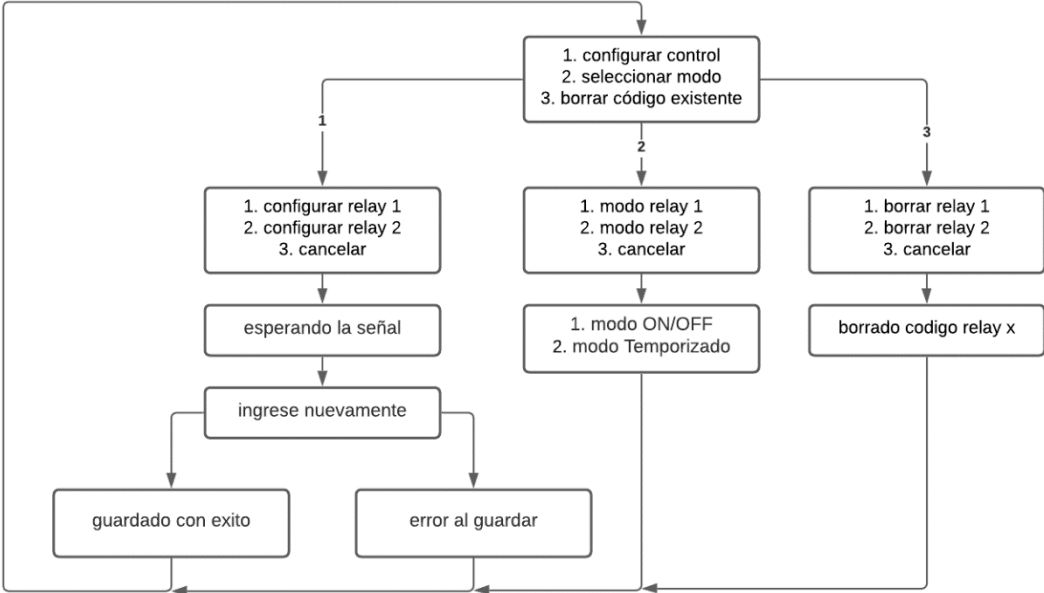


Fig. 3. Estructura general de operación de la interfaz de usuario.

Cuando alguno de los relés está configurado en modo “temporizado”, el programa desarrollado activa el temporizador Timer 1 que posee el microcontrolador, lo configura para operar en modo normal y hace uso de su interrupción por desbordamiento para implementar la temporización

requerida. De esta forma, utilizando un valor de precarga VPC en el Timer 1, con la Ec. (3) puede obtenerse el tiempo que demora en desbordarse este temporizador y producirse cada interrupción.

$$T = \frac{(2^{16} - VPC) \cdot PRE}{F_{CLK}} \quad (3)$$

donde F_{CLK} es la frecuencia de la señal de reloj del microcontrolador y PRE es el valor del prescaler usado en el Timer 1. Siendo $F_{CLK} = 16$ MHz, $VPC = 49911$ y $PRE = 1024$, las interrupciones en el Timer 1 se producen cada $T = 1$ s. Con esto el programa cuenta con una base de tiempo de 1s la cual es utilizada en la temporización de los relés.

Cuando el sistema recibe la señal que comanda a uno de los relés y este opera en modo “temporizador”, el mismo se enciende y son activadas las interrupciones del Timer 1, produciéndose estas cada 1s de acuerdo a lo mencionado. Con cada interrupción, el programa disminuye en una unidad el valor numérico obtenido en la conversión del ADC. Esta disminución continúa hasta llegar a 0 y cuando esto sucede el relé encendido durante la temporización, se apaga. Para este modo de operación se utilizó un tiempo máximo de 10s, a modo de poder comprobar su correcto funcionamiento. En caso de requerirse una temporización máxima mayor, en el código del programa debe modificarse el valor definido como “Tmax”, ubicado al comienzo del código.

Debido a que el código de las teclas programadas y el modo de operación de los relés, deben mantenerse ante eventuales cortes de energía en el circuito, una vez que el usuario ha realizado esta operación configuración del sistema el programa almacena la información correspondiente en la memoria EEPROM del microcontrolador. Esta información es recuperada de dicha memoria cada vez que se inicia la rutina principal del programa, cuando el circuito es energizado. Esto puede apreciarse en la Fig. 4.

En el programa desarrollado, la rutina principal está constantemente comprobando el estado de los pulsadores de la interfaz de usuario. En caso de que uno de ellos sea pulsado, la rutina llama a la correspondiente función. En caso de que no se accione ningún pulsador, la rutina principal llama a la función “comprobar”, la cual se encarga de comprobar la presencia de alguna señal en el puerto del microcontrolador que está conectado al receptor TL 1838. En caso de que en el puerto se detecte un estado alto, el programa comienza a analizar la señal recibida (como se explicó antes), almacenado el código en la memoria volátil. Luego de recibir todo el código el programa comprueba que el mismo sea igual a alguno de los códigos almacenados para los relés. En caso de cumplirse la igualdad, el programa comprueba el modo de operación configurado para el relé en cuestión y genera el estado lógico en el puerto de salida del microcontrolador, para comandarlo en consecuencia. Si el relé opera en modo “on/off” la acción generada es conmutar el estado lógico de la salida, en cambio si está configurado en modo “temporizado”, la salida en cuestión pasa a un estado alto que es mantenido durante el tiempo programado.

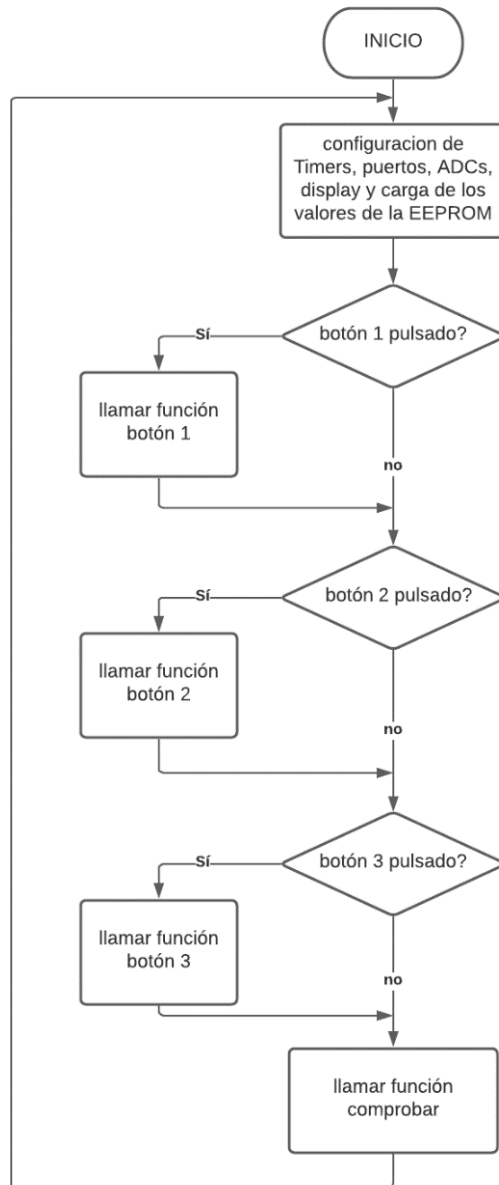


Fig. 4. Diagrama de flujos de la rutina principal del programa.

En la Fig. 4, la función “botón 1” es la que se encarga de almacenar los valores en el caso de que se desea programar las teclas de un nuevo control remoto. La función “botón 2” se encarga de seleccionar los modos de operación de los relés: Mientras que la función “botón 3” es la encargada de borrar los códigos almacenados, en caso de que ya no se desea utilizar algún control remoto programado con anterioridad.

2.2 Etapa de potencia del circuito

El circuito completo del sistema desarrollado, está en el Anexo 1, al final de este documento. En este circuito puede apreciarse que los potenciómetros usados para configurar las temporizaciones tienen sus respectivos capacitores de filtro y estabilización de la tensión analógica que se inyecta a cada canal del ADC que posee el microcontrolador en la placa Arduino Mega. El receptor infrarrojo TL 1838 incorpora el filtro RC en su salida, como lo indica el fabricante en su hoja de datos. Por otra parte, los tres pulsadores usados en la interfaz de usuario del sistema, posee resistencias pull-up para fijar el potencial en las entradas del microcontrolador sobre las cuales actúan.

En cuanto a la etapa utilizada para la activación de cada relé, el sistema propuesto utiliza el circuito mostrado en la Fig. 5. Los transistores utilizados para accionar a los relés, están conectados en configuración Darlington a modo de obtener una mayor amplificación de corriente y así los puertos del microcontrolador usados pueden operar con una baja corriente de salida. Para comandar a cada relé, el respectivo puerto es conectado a la resistencia de base R_B indicada en la Fig. 5.

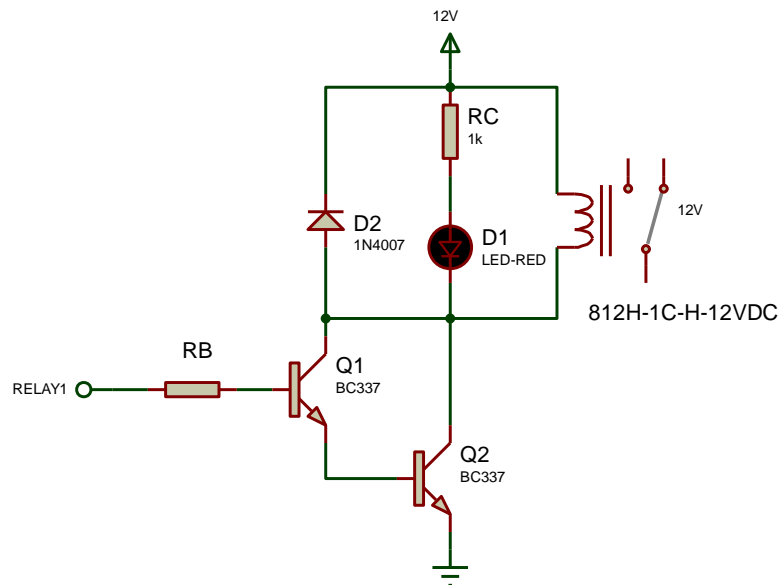


Fig. 5. Etapa buffer para el comando de los relés.

Para calcular la resistencia limitadora de corriente para el LED se utiliza la siguiente ecuación

$$RC = \frac{V_{CC} - V_{LED} - V_{CE(sat)}}{I_{LED}} \quad (4)$$

donde V_{CC} es la tensión de alimentación, V_{LED} la caída de tensión del LED y $V_{CE(sat)}$ la caída de tensión colector-emisor de saturación. Considerando $V_{CC} = 12V$, $V_{LED} = 2V$, $I_{LED} = 10 \text{ mA}$ y $V_{CE(sat)} = 0,7 \text{ V}$ [12], a través de la Ec. (4) el valor comercial de la resistencia limitadora es $RC = 1k\Omega$.

Con el valor comercial de RC y la Ec. (4) puede recalcularse la corriente a través del LED para determinar su potencia mediante la siguiente ecuación:

$$P_{RC} \geq F_S \cdot I_{LED}^2 \cdot RC \quad (5)$$

donde F_S corresponde a un factor de seguridad para contemplar cuestiones de disipación de calor en el lugar de montaje. Con $F_S = 1,5$ resulta $RC = 1k\Omega / 1/4W$.

La corriente de colector del transistor Q2 indicado en la Fig. 5, puede obtenerse en forma aproximada sumando la corriente del LED con la del relé. Considerando que la corriente para el relé utilizado es de 30 mA [13], la corriente en el colector del transistor Q1 es $I_C = 40$ mA. Con esta corriente puede verificarse la potencia disipada en el transistor mencionado, la cual está dada por:

$$P_C = I_C \cdot V_{CE(sat)} \quad (6)$$

Utilizando $V_{CE(sat)} = 0,7$ V, la potencia que disipada por el transistor es de 28 mW, inferior a 625 mW la potencia máxima de operación dada por el fabricante en la hoja de datos [12].

A partir de la corriente I_C puede obtenerse la corriente de base I_B necesaria para saturar a los transistores. Las relaciones que entre ambas corrientes en el circuito de la Fig. 5, está dada por:

$$I_B = 10 \frac{I_C}{(h_{fe\min})^2} \quad (7)$$

Considerando $h_{fe\min} = 100$, y la corriente de colector I_C que debe manejar cada transistor, la corriente de base para la saturación de los transistores Q1 y Q2 resulta en $I_B = 40$ μ A. Siendo esta corriente bastante inferior a la corriente de salida que puede manejar cada puerto del ATmega 2560 [13]. Con la corriente de base de saturación I_B , la resistencia de base RB puede dimensionarse con la siguiente ecuación:

$$RB = \frac{(V_{OH\min} - 2 \cdot V_{BE(on)\max})}{I_B} \quad (8)$$

donde $V_{OH\min}$ es la tensión de salida en estado alto que puede entregar la salida del microcontrolador y $V_{BE(on)\max}$ es la tensión base-emisor de encendido que posee el transistor Q2. Considerando $V_{OH\min} = 4,2V$ y $V_{BE(on)\max} = 1,2V$, para la corriente $I_B = 40$ μ A se obtiene $RB = 39$ $k\Omega / 1/4W$.

3. Resultados

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. Por tal motivo, el circuito del sistema fue implementado a modo de prototipo experimental, utilizándose los materiales disponibles.

Una vez dimensionados los componentes, el circuito mostrado en el Anexo 1 se ha montado en una placa de prueba (protoboard) como lo muestra la Fig. 6.

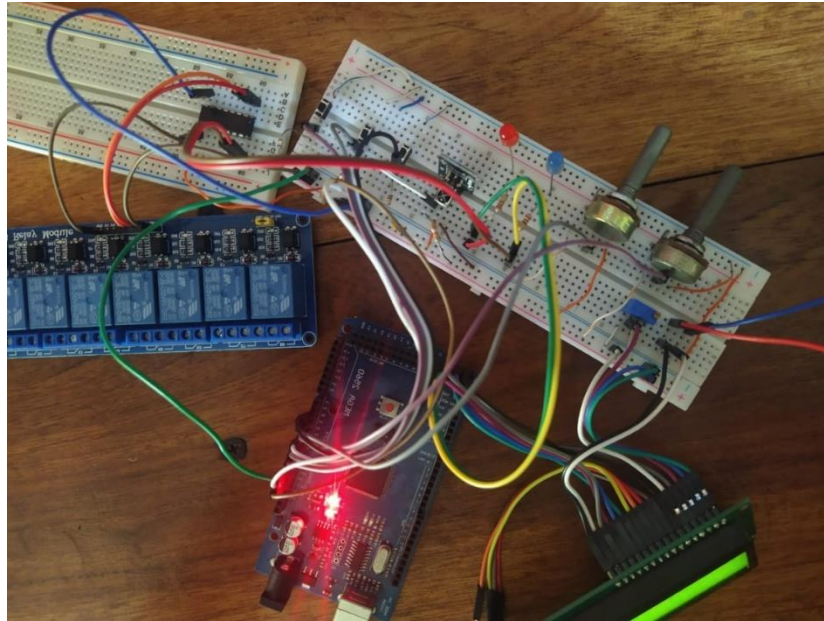


Fig. 6. Prototipo experimental del sistema para mando de relés a distancia.

En la Fig. 6 puede observarse el Arduino Mega conectado al display LCD, a los pulsadores, LEDs y los potenciómetros con los cuales se ajusta el temporizado. Todos estos elementos montados sobre el protoboard. Los LEDs están conectados a la etapa de potencia donde se encuentran los relés. Para estos últimos, por motivos de disponibilidad se utilizaron dos relés del módulo relé de 6 canales que muestra la Fig. 6. Luego de realizar este montaje, el programa desarrollado fue transferido a la memoria del ATmega2560 que posee la placa Arduino Mega.

Dado que el display LCD disponible no cuenta con suficientes caracteres como para poder mostrar el mensaje completo del menú principal, se optó por mostrar únicamente las palabras claves para entender las opciones disponibles en la configuración del sistema. Esto se muestra en la Fig. 7 (a).

Una vez encendido el sistema, para comprobar el funcionamiento del mismo fueron programadas las teclas de distintos controles remotos que estaban a disposición (Aire acondicionado, TV Philips, control universal, DIRECTV). Con esto se comprobó que el circuito respondía bien a las órdenes de las teclas programadas, encendiéndose correctamente los relés y sus LEDs asociados, tal como puede observarse en la Fig. 7 (b).

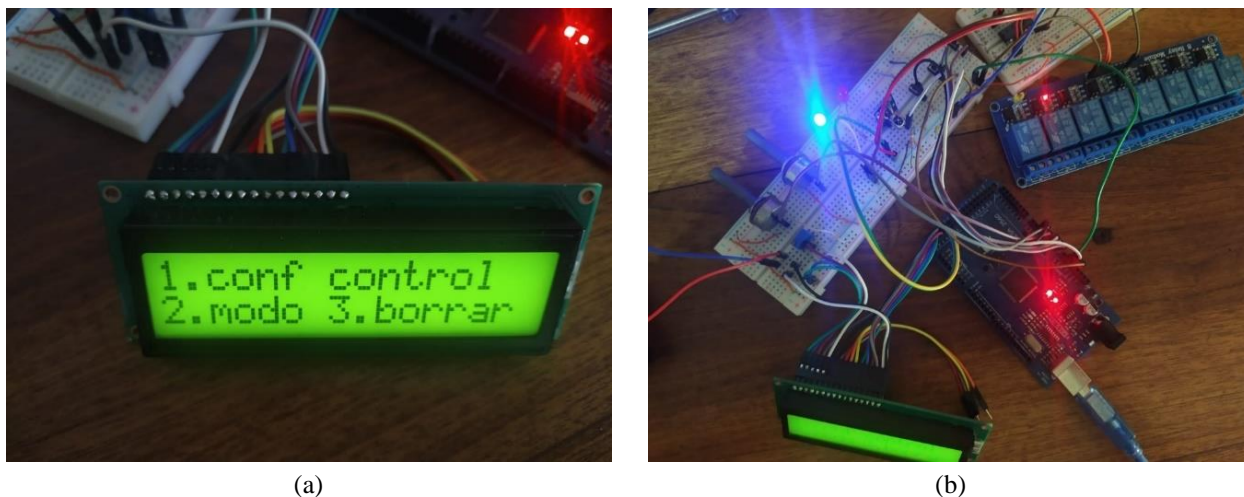


Fig. 7. (a) Menú principal mostrado en el display; (b) Prueba de funcionamiento del sistema.

También fue comprobada la operación de los relés en el modo “temporizado”. Para esto se utilizó un cronometro para comprobar si la temporización de los relés podía operar entre 1 y 10 segundos. Los resultados obtenidos en estos ensayos fueron satisfactorios.

Para comprobar la permanencia de la información almacenada en la memoria EEPROM del microcontrolador, el sistema fue desconectado y reconectado de la fuente de alimentación, reiteradas veces. A partir de esto pudo observarse que efectivamente se habían almacenado los códigos en la memoria no volátil. Otra prueba que se hizo fue el borrado de los códigos pre-grabados en la memoria no volátil, lo cual también ha funcionado bien.

4. Conclusiones

Mediante la realización de este trabajo pudo diseñarse e implementarse un sistema capaz de reconocer la mayoría de las señales emitidas por controles remotos que funcionan con infrarrojo, exceptuando algunos protocolos donde no se envían señales en pulsos de diferentes duraciones que modulan a una portadora de 38 kHz, sino que se transmite toda la información en unos cuantos nanosegundos. En estos casos el sistema desarrollado no es capaz de reconocer las diferentes teclas que posee el control remoto, dado que el *firmware* está diseñado para que guarde el código y luego reconozca el mismo código. Otro caso donde el sistema desarrollado no puede reconocer el código enviado, es cuando se envían señales parecidas, pero con algunas variantes. Como es el caso de los controles remotos de marca Sony, los cuales utilizan un bit de “palanca” que va intercambiando de estado cada vez que se vuelve a presionar la misma tecla del control, esto para que el receptor pueda comprobar si se continúa presionando la tecla correspondiente.

Por otra parte, como durante la operación normal (no durante la programación) se ignoran los métodos de comprobación de error de los distintos protocolos, puede suceder que el código enviado difiera del que se recibe el sistema y de esta forma el programa desarrollado no sería capaz de reconocer la tecla presionada. Si bien el programa desarrollado permite utilizar la mayoría de los controles remotos para comandar a los relés, en caso de necesitarse un sistema más robusto, donde el

protocolo es bien conocido, será más conveniente decodificar la señal y realizar un algoritmo capaz de hacer las comprobaciones de error que se requieren en el protocolo utilizado.

El sistema propuesto en este trabajo fue implementando satisfactoriamente pudiendo cumplirse con los objetivos del trabajo integrador propuesto en la asignatura Técnicas Digitales 2. Con el desarrollo del mismo se pudieron afianzar y profundizar los saberes adquiridos en distintas asignaturas cursadas de la carrera. Estos saberes involucraron a los circuitos y sistemas digitales, donde fue necesario usar microcontroladores y comprender la utilidad de sus capacidades tales como las salidas PWM, los conversores ADC y la programación de estos dispositivos. Los saberes involucrados también tuvieron que ver con la electrónica analógica, ya que en el desarrollo del trabajo se aplicaron conocimientos sobre componentes electrónicos tales como transistores, LEDs, etapas de potencia, disipación de calor en los componentes, adquiridos estos en materias tales como “Electrónica Analógica”. Por otra parte, el trabajo desarrollado permitió el acercamiento de nuevos conocimientos, como los referentes a la modulación de las señales, que serán abordados en futuras asignaturas de la carrera (por ejemplo, en Sistemas de Comunicaciones). También este trabajo propició la investigación en los temas de los cuales se tenía desconocimiento, como es el caso de los protocolos utilizados por los controles remotos.

Referencias

[1]	«Diferencias entre Control Remoto IR vs RF » [En línea]. Available: https://t.ly/fZQO . [Último acceso: 28-07-2021].
[2]	«RF vs IR. Qué control remoto es mejor» [En línea]. Available: https://t.ly/zNJD . [Último acceso: 04-08-2021].
[3]	Microchip, ATmega2560 Datasheet. [En línea]. Available: https://t.ly/WgHr [Último acceso: 05-08-2021].
[4]	«NEC Protocol - SB-Projects » [En línea]. Available: t.ly/wT1b . [Último acceso: 28-07-2021].
[5]	«JVC Protocol - SB-Projects » [En línea]. Available: t.ly/h0xF . [Último acceso: 28-07-2021].
[6]	«Nokia NRC17 Protocol - SB-Projects » [En línea]. Available: t.ly/mouR . [Último acceso: 28-07-2021].
[7]	«Philips RC-5 Protocol - SB-Projects » [En línea]. Available: t.ly/NoJ5 . [Último acceso: 28-07-2021].
[8]	«Sony SIRC - SB-Projects » [En línea]. Available: t.ly/vdTx . [Último acceso: 28-07-2021].
[9]	«TL 1838 datasheet» [En línea]. Available: https://t.ly/iLck . [Último acceso: 05-08-2021].
[10]	«AVR Programación en C» [En línea]. Available: https://acortar.link/q5bQbp [Último acceso: 05-08-2021].
[11]	«Display LCD 16x2 JHD659 datasheet,» [En línea]. Available: t.ly/keSa . [Último acceso: 28 07 2021].
[12]	«BC337 Datasheet (PDF) - Fairchild Semiconductor,» [En línea]. Available: t.ly/Cn3f . [Último acceso: 28 07 2021].
[13]	«Relé 812H-1C-C Datasheet» [En línea]. Available: t.ly/Bx64 . [Último acceso: 28 07 2021].
[14]	«TL1838 Datasheet (PDF) » [En línea]. Available: t.ly/X6W0 . [Último acceso: 28 07 2021].
[15]	«IR Remote Control Theory - SB-Projects » [En línea]. Available: t.ly/pGJb . [Último acceso: 28 07 2021].

Anexo 1: Circuito del sistema para mando de relés a distancia

