Enviado: 01/01/2001 *Recibido: 01/02/2001* *Aceptado: 01/03/2001*

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

VISIÓN ELECTRÓNICA

Algo más que un estado sólido

https://doi.org/10.14483/issn.2248-4728

VISIÓN ELECTRÓNICA

PROTOBOARD CONFIGURABLE POR SOFTWARE, DENOMINADA "SOFTBOARD"

SOFTWARE CONFIGURABLE BREADBOARD, CALLED "SOFTBOARD"

Edwin Saúl González Contreras [1], William Camilo Jiménez Rodríguez [2]

Resumen

Distintos métodos de prototipado de circuitos electrónicos se han desarrollado a lo largo del último siglo, donde la experiencia ha seleccionado el proceso más sencillo, económico, práctico y reutilizable, la protoboard, en ella es posible realizar el montaje de circuitos en un corto tiempo y ponerlos a prueba, además, su contribución al sector académico es base de la experticia de un profesional a fin. El presente documento muestra los resultados del proceso de diseño, desarrollo y construcción de la herramienta "Softboard", la cual pone a disposición de los estudiantes y profesionales, una alternativa para el montaje de circuitos electrónicos apoyados en un software que comunica a un microcontrolador la interconexión entre filas y/o columnas de una protoboard por medio de relés de estado sólido. Aquí se describen características eléctricas del dispositivo, diseño del prototipo, costos del proyecto, comparación de funcionamiento con una protoboard tradicional y algoritmos de programación.

Palabras clave: Protoboard, relé de estado sólido, software, prototipos electrónicos, "Softboard".

[1] Estudiante de tecnología en electrónica industrial.: Universidad Distrital Francisco José de Caldas. E-mail: esgonzalezc@correo.udistrital.edu.co

[2] Estudiante de tecnología en electrónica industrial.: Universidad Distrital Francisco José de Caldas. E-mail: wcjimenezr@correo.udistrital.edu.co

Abstract

Different methods of prototyping electronic circuits have been developed over the last century, where experience has selected the simplest, most economical, practical and reusable process, the breadboard, in which it is possible to assemble circuits in a short time and putting them to the test, in addition, their contribution to the academic sector is the basis of the expertise of a professional. This document shows the results of the process of design, development and construction of the “Softboard” tool, which makes available to students and professionals an alternative for the assembly of electronic circuits supported by software that communicates to a microcontroller the interconnection between rows and / or columns of a breadboard by means of solid state relays. This describes the electrical characteristics of the device, prototype design, project costs, performance comparison with a traditional breadboard, and programming algorithms.

Keywords: Breadboard, Solid State Relay, software, electronic prototypes, “Softboard”

1. Introducción

Desde antes de la revolución dada con la invención del transistor, han sido necesarias herramientas que ayuden a los diseñadores a elaborar prototipos de circuitos eléctricos y electrónicos, después de los años 1900 se implementaron variadas técnicas y métodos [1].

Con esto, todos los días se diseñan circuitos electrónicos para resolver problemas en ambiente académico o industrial, los softwares de simulación resuelven problemas mediante la emulación de los componentes electrónicos interconectados con otros, este proceso permite al diseñador hacer todas sus pruebas y obtener un diseño funcional [2]. Aun así, la simulación se complementa con el montaje físico para corroborar el funcionamiento del circuito, aquí es donde suelen haber conexiones erróneas, cables, conectores que generan inestabilidad o

ruido en el circuito, además, gran parte del tiempo en el laboratorio es dedicado a realizar la conexión de los elementos [3].

Como señala [4], “El aprendizaje por proyectos es una opción metodológica basada en la investigación, cuyo objetivo es organizar los contenidos curriculares bajo un enfoque globalizador y significativo, relacionando los conocimientos escolares con los de la vida cotidiana”, lo que permite a los estudiantes relacionarse con problemas similares a los que encontrarán en su vida laboral. En las áreas de formación, que tienen que ver con la electrónica, es común encontrar temáticas integradoras aplicadas al desarrollo de prototipos electrónicos, los cuales conllevan a un largo proceso de desarrollo, en donde los estudiantes deben incorporar los conocimientos adquiridos para alcanzar los objetivos de formación [5]. En la mayoría de estos proyectos los estudiantes requieren el montaje de un circuito electrónico con el fin de complementar la formación teórica y verificar el funcionamiento del circuito [6].

El desarrollo de un nuevo dispositivo para prototipado de circuitos electrónicos denominado “Softboard” soluciona problemas cotidianos del entorno de laboratorio, aumenta la eficiencia y tecnifica el proceso con integración de hardware y software.

A continuación, se explican los pasos que condujeron al diseño y construcción del primer modelo de “Softboard”, iniciando por la selección del interruptor controlado por voltaje donde se tuvo en cuenta características eléctricas convenientes para el reemplazo de un cable y disponibilidad del mercado internacional. Posteriormente se seleccionó una lista de circuitos electrónicos fundamentales, los cuales pasaron por un proceso de simulación con conexiones directas y conexiones mediante relés de estado sólido (SSR), comprobada su funcionalidad se procedió a realizar el diseño de software (GUI) mediante el lenguaje de programación Python, comunicado por RS-232 a un microcontrolador encargado de controlar los SSR. Teniendo completa la parte de simulaciones y algoritmos se siguió al diseño de una tarjeta de circuitos impresos (PCB) adaptada a una protoboard tradicional, con la construcción del prototipo

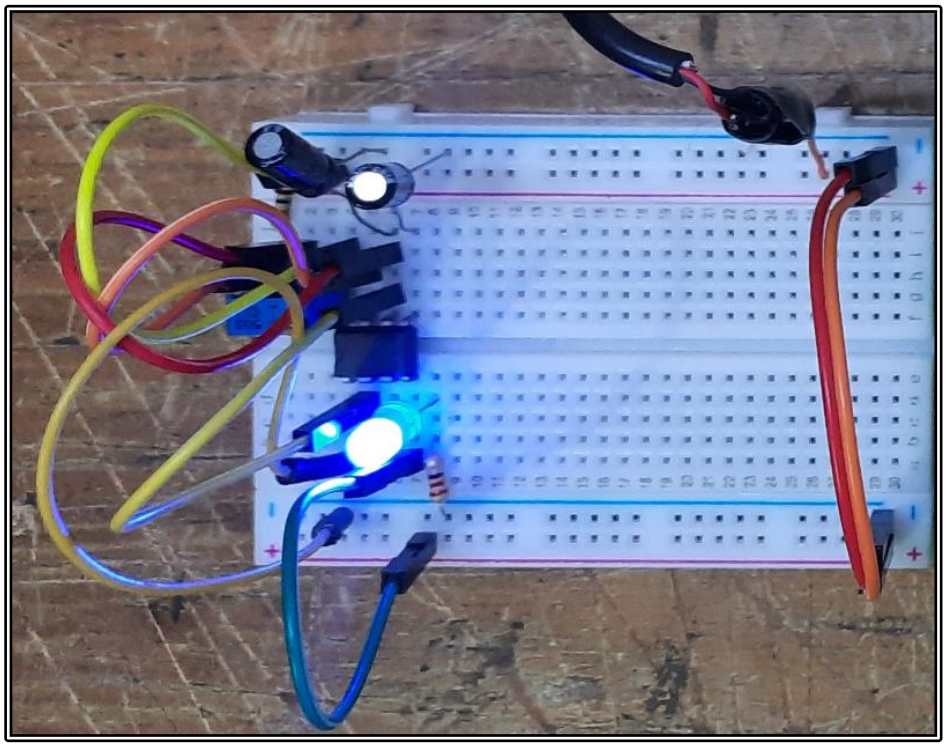
finalizada se realizó el montaje de los circuitos seleccionados en una protoboard y una “Softboard”, donde se comparan una serie de criterios que permiten definir la viabilidad electrónica del dispositivo, y finalmente se presentan discusiones acerca de la comodidad, tiempo, limitaciones y los costos derivados de su construcción.

2. Marco de referencia

2.1. Estado del arte

La protoboard consiste en un tablero con orificios que se encuentran conectados eléctricamente entre sí. Los orificios permiten el montaje de componentes electrónicos y cables para interconectarlos. La placa está hecha de un material aislante y un material conductor [7].

Figura 1. Montaje de un circuito en protoboard

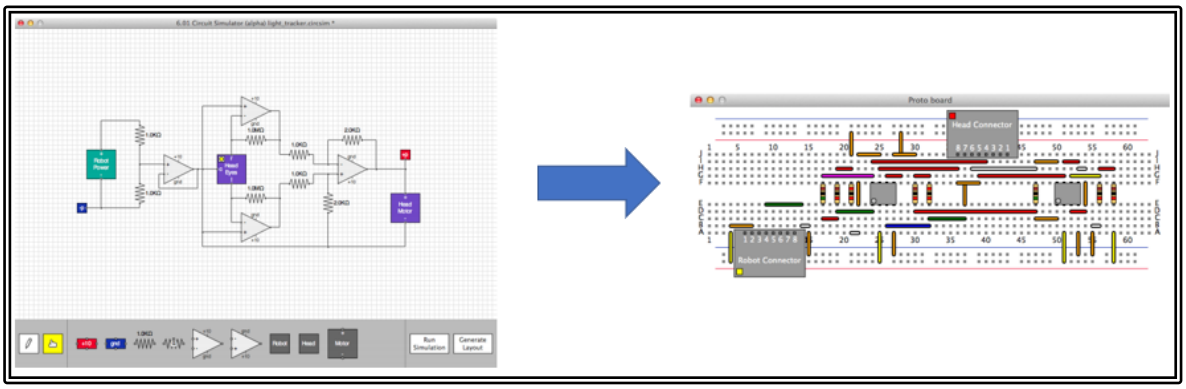


Fuente: elaboración propia

Tomando ventaja de su conveniente diseño y distribución entre aislantes y conductores, la protoboard permite ubicar componentes electrónicos de manera temporal, así también permite

la conexión entre ellos por medio de cables o conectores. Se tienen herramientas de software que ayudan al prototipado, es el caso de “Automatic Protoboard Layout from Circuit Schematics” [3], un software que, a partir de una simulación, se aplica un algoritmo que indica cómo se debe ubicar cada componente y su conexión con otros sobre la protoboard.

Figura 2. Programa para distribución y conexiones en una protoboard.

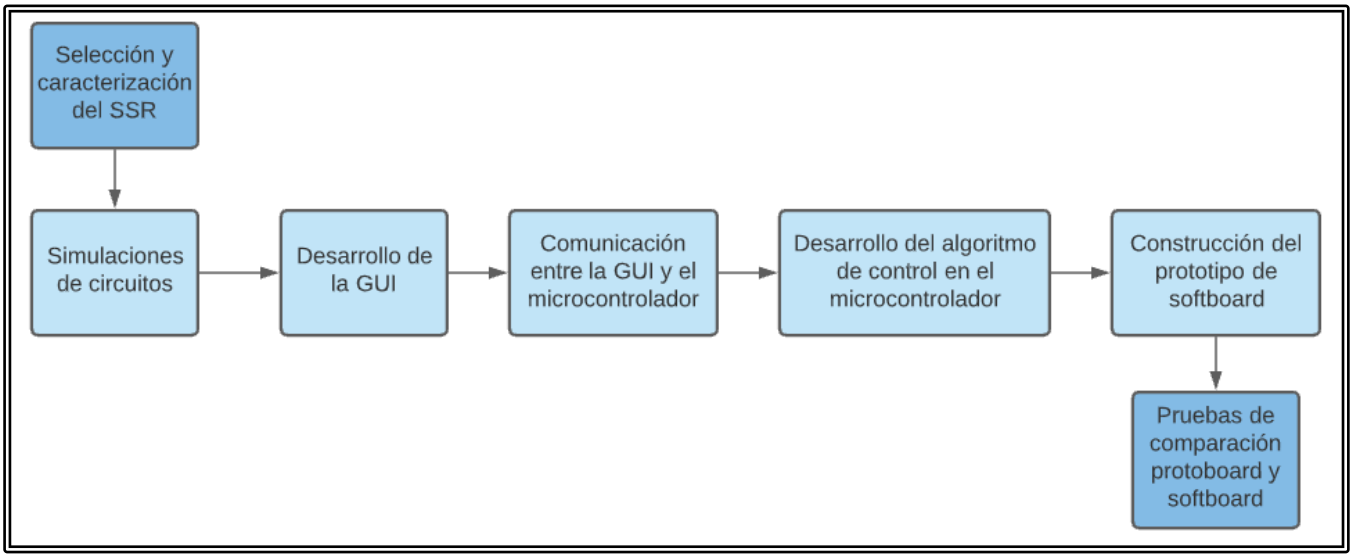


Fuente: “Automatic Protoboard Layout from Circuit Schematics” [3]

3. Metodología y materiales

Se siguió la metodología presentada en la figura 3, para el desarrollo del proyecto.

Figura 3. Diagrama de flujo de la metodología



Fuente: elaboración propia

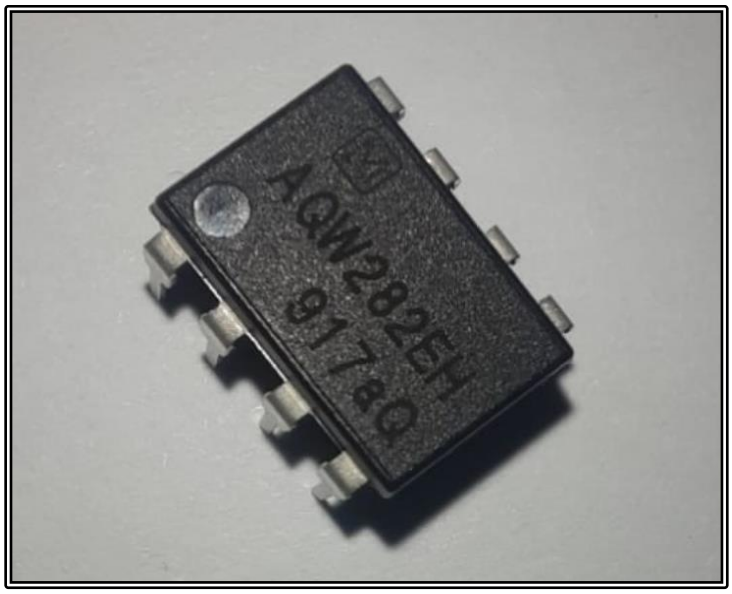
3.1. Caracterización del relé de estado sólido AQW282EH

Para la selección del interruptor se tuvo en cuenta las siguientes características:

- Encapsulado de agujero pasante
- Voltaje de carga
- Corriente de carga
- Voltaje de aislamiento
- Trabajo en DC/AC
- Resistencia de salida
- Capacitancia de salida

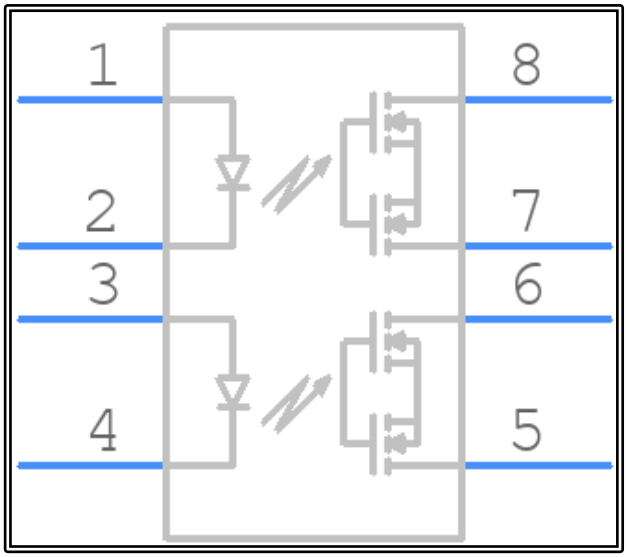
Para lo cual se determinó la referencia AQW282EH, un SSR de la fábrica Panasonic de la línea "*PhotoMOS Relays"*, usos típicos en: modem sensores, equipos de comunicaciones y seguridad. En la figura 5 se puede observar el diagrama de conexiones del integrado, que está compuesto por dos interruptores tipo MOSFET que son activados mediante un diodo emisor de luz, lo que permite aislar la parte de control de la parte de potencia.

Figura 4. Circuito integrado AQW282EH



Fuente: elaboración propia

Figura 5. Circuito integrado AQW282EH, diagrama de conexiones



Fuente: ficha técnica AQW282EH

Las características más relevantes para este integrado son:

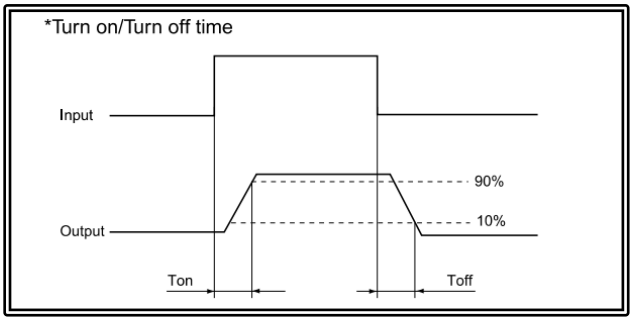
Tabla 1. Principales características del AQW282EH

Características	AQW282EH
Corriente de carga	400mA
Voltaje de carga	60V
Encapsulado	DIP 8P
Voltaje de aislamiento	5000V
Trabajo DC/AC	DC/AC
Resistencia de salida (máx)	0.8Ω -2,5Ω
Capacitancia de salida	1.5 pF
Tiempo de encendido (TON)	1.8ms—5ms
Tiempo de apagado (TOFF)	0.5ms—2ms

Fuente: elaboración propia

En la tabla 1 se observa el tiempo de encendido del interruptor TON y el tiempo de desactivación TOFF, cuyos valores máximos son 5ms y 2ms respectivamente. En la figura 6 se representa la gráfica de comportamiento del tiempo de encendido TON, que corresponde al tiempo que toma abrir al 90% el canal y el TOFF corresponde al tiempo que tarda después de poner en cero la señal de entrada y cerrar el canal hasta el 10%.

Figura 6. Diagrama TON y TOFF

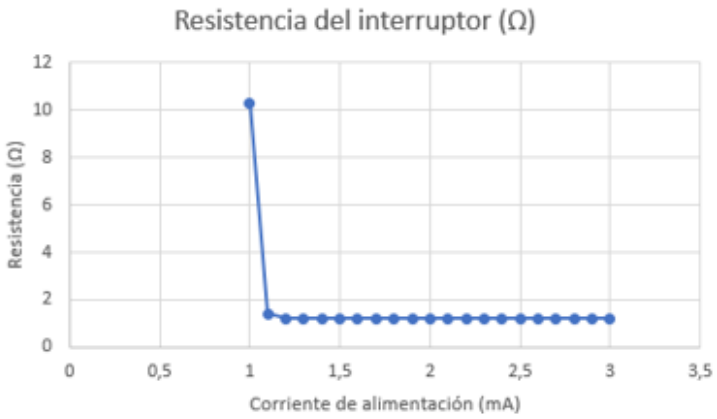


Fuente: ficha técnica AQW282EH

En la tabla 1 obtenemos los valores de resistencia para el canal, varían entre los 0.8Ω a 2.5Ω, para comprobar esta información en tabla 2 se encuentran los resultados de las pruebas realizadas, estas se ejecutaron a temperatura ambiente (25° Celsius), con rangos de corriente de alimentación entre 1mA y 3mA.

Tabla 2. Pruebas resistencia de salida, variación de corriente de alimentación

Corriente de alimentación (mA)	Resistencia del interruptor (Ω)
1	10,3
1,1	1,4
1,2	1,2
1,3	1,2
1,4	1,2
1,5	1,2
1,6	1,2
1,7	1,2
1,8	1,2
1,9	1,2
2	1,2
2,1	1,2
2,2	1,2
2,3	1,2
2,4	1,2
2,5	1,2
2,6	1,2
2,7	1,2
2,8	1,2
2,9	1,2
3	1,2



Fuente: elaboración propia

Se observa un valor de 10.3Ω con alimentación de 1mA, esto se debe a que es un interruptor normalmente abierto por lo que viene de un valor de resistencia infinita, luego, se estabiliza en 1.2Ω aplicando la corriente recomendada por el fabricante.

En la tabla 3 se analiza el comportamiento de resistencia en el interruptor variando la corriente de carga entre 318mA-1mA, se mide voltaje del interruptor y corriente total del circuito para aplicar la ley de Ohm y obtener su valor en ohmios.

Tabla 3. Pruebas de corriente de carga

Corriente (mA)	Voltaje SSR (mV)	Resistencia (Ω)
318	363	1,14
300	350	1,17
250	290	1,16
200	230	1,15
150	170	1,13
100	113	1,13
50	56	1,12
25	28	1,12
20	22,3	1,12
15	16,7	1,11
10	11	1,10
5	5,5	1,10
1	1,1	1,10

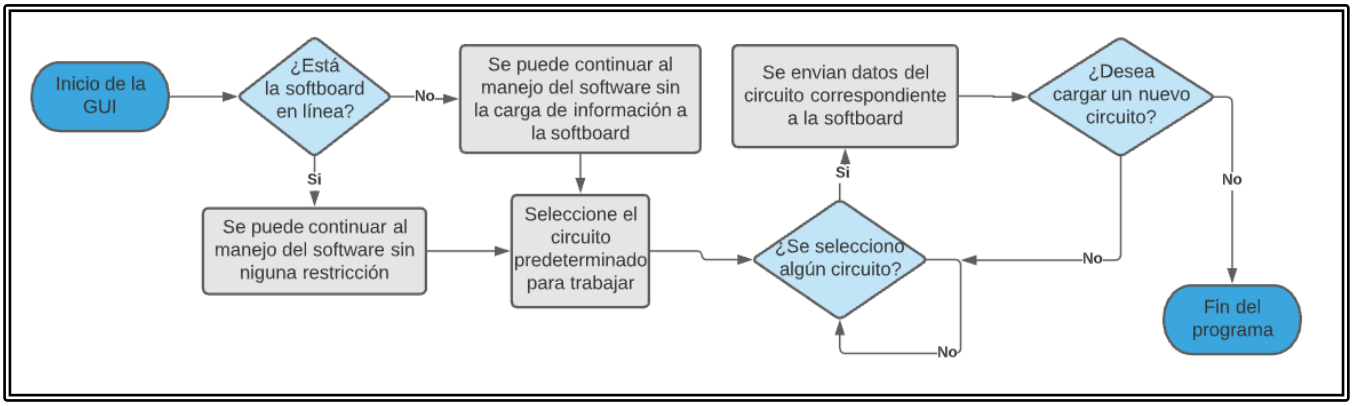
Fuente: Elaboración propia

Se analizamos las medidas extremas encontramos un cambio decreciente de 40mΩ, desde que se aplicó 318mA hasta 1mA, la resistencia se mantiene entre 1.14Ω y 1.10Ω.

3.2. Desarrollo de la GUI

La interfaz se desarrolló en el lenguaje de programación Python con el IDE Spyder, el manejo es práctico, como muestra el siguiente flujograma:

Figura 7. Flujograma de la GUI Fuente: Autor



Fuente: elaboración propia

El software se diseñó con la librería pygame [8], con esta se realizaron interacciones intuitivas que permiten al usuario proceder a seleccionar un circuito predeterminado, en la figura 8 se presenta un fragmento del código que dibuja la softboard.

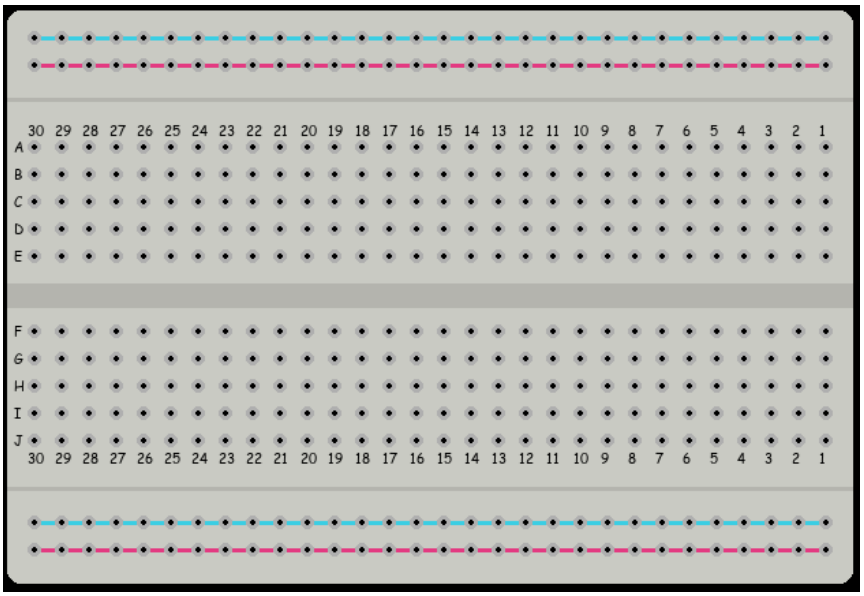
Figura 8. Fragmento de código GUI

```
pygame.draw.rect(pantalla,NEGRO,[0,0,1200,600],0,10) #Fondo
pygame.draw.rect(pantalla,PB,[80,50,620,420],0,10) # Base PB
pygame.draw.line(pantalla,AZULM, [100,70],[680,70], 3) #Negativa superior
pygame.draw.line(pantalla,ROJO, [100,90],[680,90], 3) #Positiva superior
pygame.draw.line(pantalla,GRIS, [80,115],[699,115], 3) #Separador superior
pygame.draw.line(pantalla,AZULM, [100,425],[680,425], 3) #Negatva inferior
pygame.draw.line(pantalla,ROJO, [100,445],[680,445], 3) #Positiva inferior
pygame.draw.line(pantalla,GRIS, [80,400],[699,400], 3) #Separador inferior
pygame.draw.line(pantalla,GRIS, [80,258],[699,258], 18) #Separador inferior

for x in range(100,700,20):  #BUS "+ -" SUPERIOR
    for y in range(70,110,20):
        pygame.draw.circle(pantalla, GRIS_CLARO, [x, y], 5)
        pygame.draw.circle(pantalla, NEGRO, [x, y], 2)

for x in range(100,700,20):  #1-30 F-G
    for y in range(150,250,20):
        pygame.draw.circle(pantalla, GRIS_CLARO, [x, y], 5)
        pygame.draw.circle(pantalla, NEGRO, [x, y], 2)

for x in range(100,700,20):  #1-30 A-E
    for y in range(285,385,20):
        pygame.draw.circle(pantalla, GRIS_CLARO, [x, y], 5)
        pygame.draw.circle(pantalla, NEGRO, [x, y], 2)

for x in range(100,700,20):  #BUS "+ -" INFERIOR
    for y in range(425,465,20):
        pygame.draw.circle(pantalla, GRIS_CLARO, [x, y], 5)
        pygame.draw.circle(pantalla, NEGRO, [x, y], 2)
ytc=357 #F-J TEXT
for r in range(0,5,1):
    text = font.render(row[r], True, (NEGRO))
    pantalla.blit(text, (85,ytc))
    ytc=ytc-20
ytc=222#A-E TEXT
for r in range(5,10,1):
    text = font.render(row[r], True, (NEGRO))
    pantalla.blit(text, (85,ytc))
    ytc=ytc-20
xtc=95
for c in range(0,30,1):#30-1 top & bottom
    text = font.render(column[c], True, (NEGRO))
    pantalla.blit(text, (xtc,130))
    pantalla.blit(text, (xtc,370))
    xtc=xtc+20
```

Fuente: elaboración propia

El resultado de este código de la figura 8, es el dibujo que se muestra en la figura 9:

Figura 9: Softboard dibujada con pygame



Fuente: elaboración propia

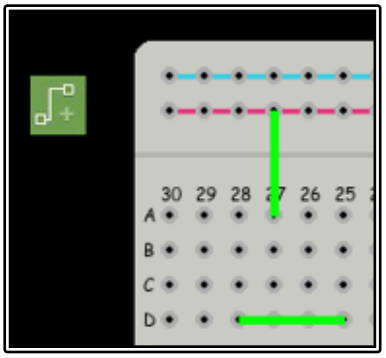
Sobre la interfaz gráfica el usuario puede interactuar al seleccionar entre 13 opciones de configuración como se muestra en la figura 10:

Figura 10. Selección de circuitos precargados

Fuente: Autor

El usuario también tiene la posibilidad de dibujar las interconexiones mediante el botón “Cable”, presentado en la figura 11:

Figura 11. Uso del botón “Cable”



Fuente: elaboración propia

La funcionalidad anterior es obtenida mediante la función “vector_sb” de la figura 12, en conjunto con la función “wire” de la figura 13.

Figura 12. Función “vector_sb”

```
def vector_sb():
    posicion_mouse()
    global y_coor,x_coor,x_en,y_en
    cont_x=0
    x_en=0
    y_en=0
    for xc in range (100,440,20):
        xc_pasada=xc-5
        xc_futura=xc+5
        cont_x=cont_x+1
        if xm<xc_futura and xm>xc_pasada:
                x_en=cont_x
                x_coor=xc
    cont_y=0
    for yc in range (70,250,20):
        yc_pasada=yc-5
        yc_futura=yc+5
        if yc==110 or yc==130:
            cont_y=cont_y
        else:
            cont_y=cont_y+1
            if ym<yc_futura and ym>yc_pasada:
                    y_en=cont_y
                    y_coor=yc
    for yc in range (285,465,20):
        yc_pasada=yc-5
        yc_futura=yc+5
        if yc==385 or yc==405:
            cont_y=cont_y
        else:
            cont_y=cont_y+1
            if ym<yc_futura and ym>yc_pasada:
                    y_en=cont_y
                    y_coor=yc
```

Fuente: elaboración propia

Figura 13. Función “wire”

```
def wire():
    global w
    mantener=True
    while mantener:
        for event in pygame.event.get():
            if event.type == pygame.MOUSEBUTTONDOWN:
                vector_sb()
                if x_en!=0 and y_en!=0:
                   e_xm.append(x_coor)
                   e_ym.append(y_coor)
            if event.type == pygame.MOUSEBUTTONUP:
                    vector_sb()
                    wires.append(1)
                    s_xm.append(x_coor)
                    s_ym.append(y_coor)
                    w=w+1
                    mantener=False
                    botones[0]['on_click']=False
                    print (s_ym,e_ym,s_xm,e_xm)
```

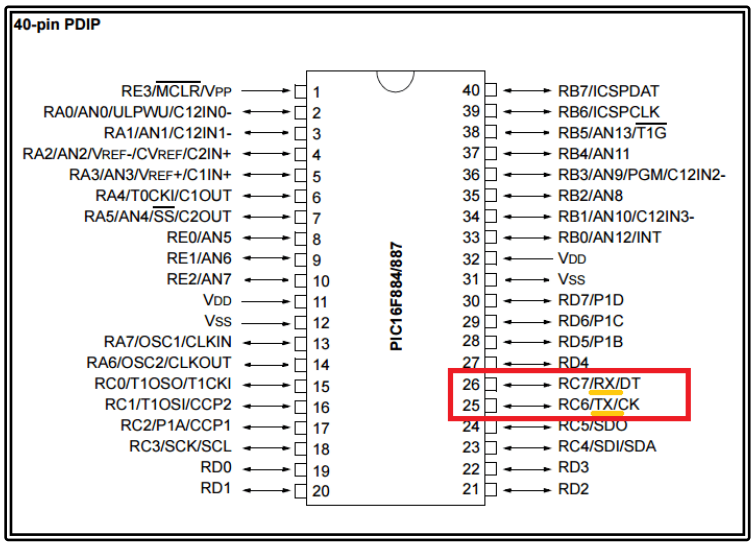
Fuente: elaboración propia

3.3. Comunicación entre la interfaz gráfica de usuario y softboard

3.3.1. Requerimientos de hardware

Para enviar y la recibir datos se implementa un protocolo de comunicación RS232 asincrónica, utilizando la UART del PIC16F877A, el pin de transmisión de datos (TX) corresponde al pin número 25 del micro y el de recepción (RX) corresponde al pin 26.

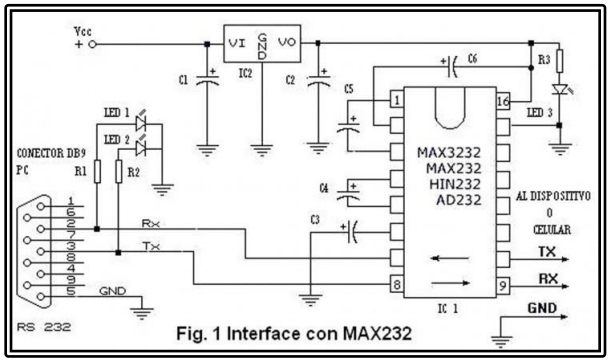
Figura 14. Ubicación RX y TX Pic16f877A



Fuente: Ficha técnica PIC16F877A

Para la comunicación con el PC se requiere el uso de Hardware adicional para convertir las señales de voltaje del Ordenador a señales compatibles TTL (Transistor–Transistor–Logic). Para esto se utilizó un módulo MAX232, cuyo esquemático se muestra en la figura 15.

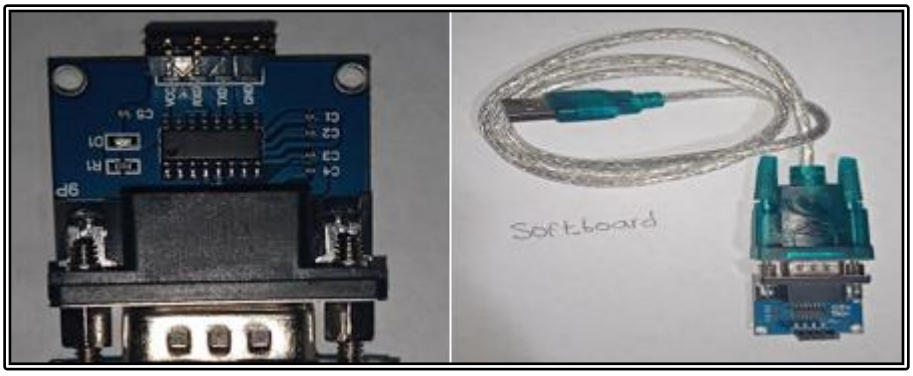
Figura 15. Circuito de conexión MAX232 y PIC16F887A



Fuente: Ficha técnica MAX232

Como la mayoría de ordenadores actuales no cuentan con un puerto RS232 físico, se utiliza un cable conversor serial-USB. El cable se conecta al módulo como se muestra en la figura 16, adicionalmente al módulo se le deben conectar los puertos VCC, TX, RX y GND del PIC.

Figura 16. Conversor serial a USB



Fuente: Ficha técnica RS232-USB

3.3.2. Software

Para establecer la comunicación serial entre el ordenador y el PIC, se realiza un software mediante el lenguaje de programación Python, haciendo uso de la librería pyserial, para el PIC se desarrolló un programa en lenguaje C utilizando el entorno de desarrollo PIC C compiler.

Código Python

Para activar o desactivar los interruptores se envía una cadena, compuesta por ‘1’(unos) o ‘0’(ceros), al microcontrolador. El ‘1’ indica que el SSR correspondiente se debe activar y el ‘0’ es para desactivarlo. Para construir la cadena que se va a enviar se desarrolló la función ‘conversión’, que recibe como parámetro un vector con los relés que se quieren activar y construye una cadena dependiendo de este vector. En la figura 17 se observa la cadena que retorna la función “conversion” para el vector [1,5,9,12,22], lo cual indica que se quiere activar los SSR número 1,5,9,12 y 22. Además de los ‘1’(unos) y ‘0’(ceros) que componen la cadena se le agrega una ‘s’ al inicio y una ‘e’ al final, indicando donde empieza y donde termina la cadena.

Figura 17. Función “conversión”

```python
def conversion(vector_activar):

    cadena=""
    reles={}

    for valor in range(1,23):
        reles[valor]=False

    reles_activar = vector_activar

    for x in reles_activar:
        reles[x]=True


    for x in reles:
        if reles[x]==False:
            cadena=cadena+"0"
        else :
            cadena=cadena+"1"
    cadena="s"+cadena+"e"
    return cadena
```

Fuente: elaboración propia

En la figura 18 se observa la cadena que retorna la función “conversión” para el vector [1,5,9,12,22].

Figura 18. Función “conversión”

```
In [123]: print(conversion([1,5,9,12,22]))
s1000100010010000000001e
```

Fuente: elaboración propia

En la figura 19 se observa el código fuente de la función “envio_datos”, la cual recibe como parámetro una cadena y la envía por el puerto serie. En la tercera línea se define el puerto serial con su respectiva configuración. La función retorna un valor booleano, ‘True’ si la comunicación se completó correctamente y ‘false’ si ocurrió algún error durante la ejecución.

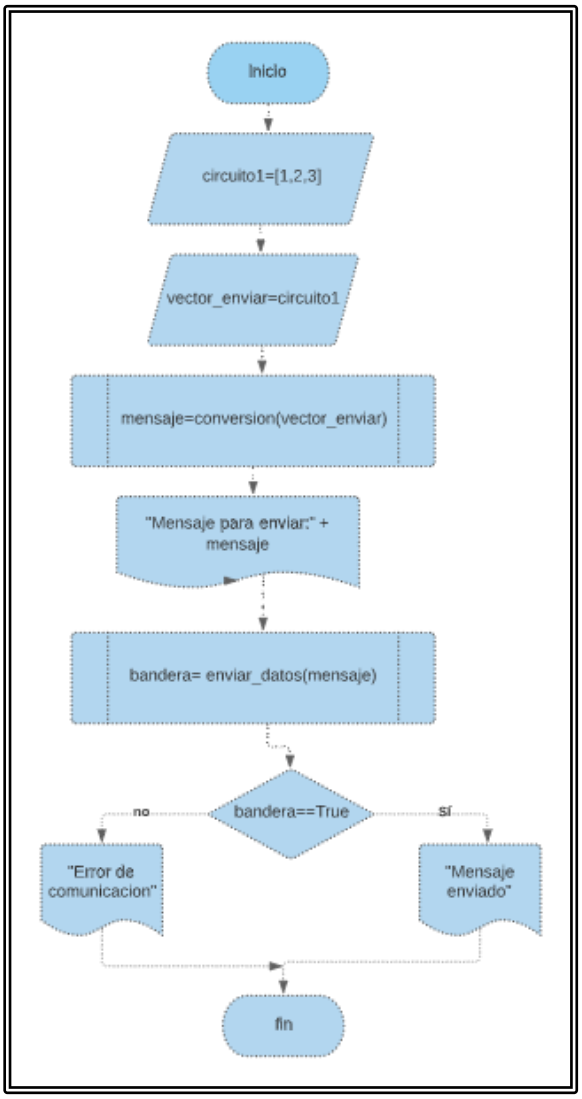
Figura 19. Función “envio_datos”

```
def envio_datos(cadena):
    try:
        com_serial = serial.Serial(port='COM10',
                                   baudrate=9600,
                                   bytesize=serial.EIGHTBITS,
                                   parity=serial.PARITY_NONE,
                                   timeout=1,
                                   write_timeout=1)
        bandera=False;
        time.sleep(0.1)
        enviar=str.encode(cadena)
        print("Cadena a enviar: ")
        print(enviar)
        com_serial.write(enviar)
        com_serial.close()
        print("Mensaje enviado")
        bandera=True
        return bandera
    except:
        print("Error en la funcion: envio_datos")
```

Fuente: elaboración propia

En la figura 20 se representa en diagrama de flujo el proceso para el envío de datos al microcontrolador. Como primera parte se establecen los interruptores que se quieren activar y se crea la cadena correspondiente y se envía con la función “envio_datos”.

Figura 21. Diagrama de flujo del algoritmo de comunicación.

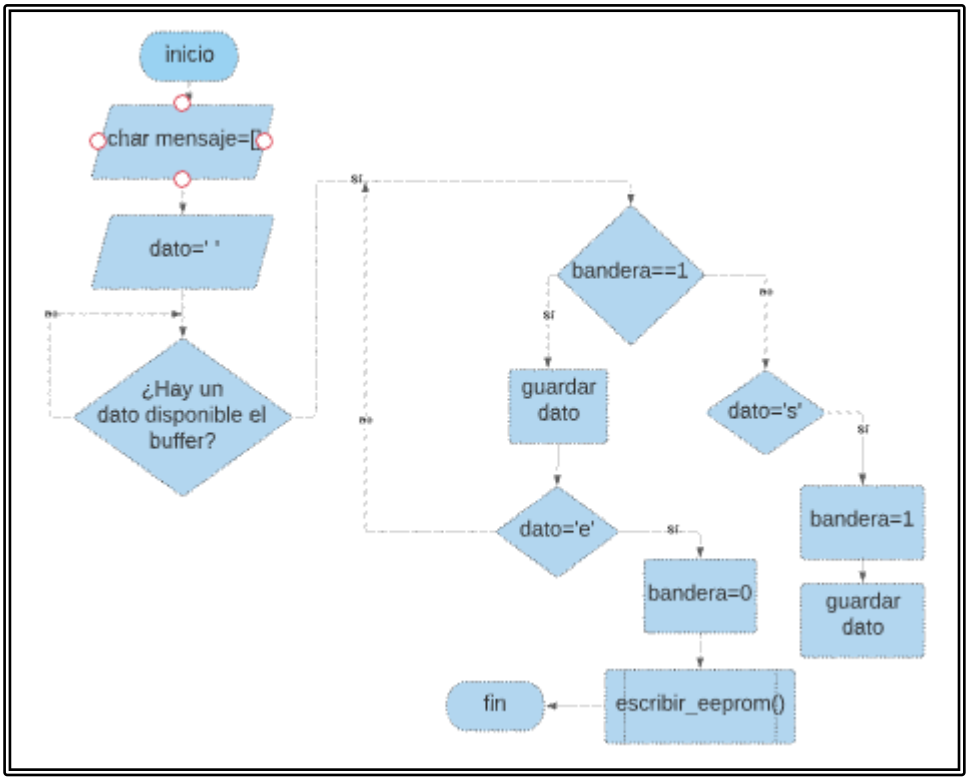


Fuente: elaboración propia

Código microcontrolador

Para la recepción de la cadena de datos, se realizó un algoritmo el cual permite almacenar la cadena en un vector tipo char. Posterior a esto se almacena en la memoria EEPROM del microcontrolador cada carácter del vector desde la dirección 0(cero) hasta la dirección de memoria 23 (veintitrés), almacenando un total de 24 bytes. En la figura 22 se muestra el diagrama de flujo con el proceso correspondiente.

Figura 22. Diagrama de flujo recepción de datos del microcontrolador



Fuente: elaboración propia

Cuando el programa detecta el final de la cadena, es decir el carácter ‘e’, se ejecuta la función “escribir_eeprom” la cual almacena en la memoria del microcontrolador la cadena de datos almacenada en la variable mensaje. El código fuente de esta función se puede observar en la figura 23, que mediante un ciclo “for” almacena en cada dirección de memoria un carácter del mensaje.

Figura 23. Función “escribir_eeprom”

```
void escribir_eeprom(){
        int cont1=0;
        int cont2=0;
        for (cont1=0; cont1<24;cont1++){
           write_eeprom(cont1,mensaje[cont2]);
           delay_ms(5);
           cont2++;
         }
```

Fuente: elaboración propia

3.5. Desarrollo del algoritmo de control en el microcontrolador

La activación o desactivación de los interruptores se realiza mediante la función “control_salidas”, cuyo código fuente se puede visualizar en la figura 24, esta función recibe como parámetro la posición de inicio de lectura EEPROM para el puerto correspondiente y la cantidad de salidas asignadas a ese puerto, estos parámetros se asignan como se muestra en la figura 25, donde un condicional evalúa si hay un registro en la memoria, esto se hace leyendo las posiciones de memoria en las que se encuentran los identificadores de inicio y fin de la cadena, ‘s’ y ‘e’ respectivamente.

Figura 24. Función “control_salidas”

```
int control_salidas(int dir_memoria, int pines_puerto){
   int valor=0;
   int valor2=0;
   int i=0;
   for (i=0; i<pines_puerto;i++){
      if (read_eeprom(dir_memoria)=='0'){
         valor2=0;
      }
      else if (read_eeprom(dir_memoria)=='1'){
         valor2=1;
      }
      valor=valor+(valor2)*pow(2,i);
      delay_ms(10);
      dir_memoria++;
   }
   return valor;
}
```

Fuente: elaboración propia

Figura 25. Asignar salidas

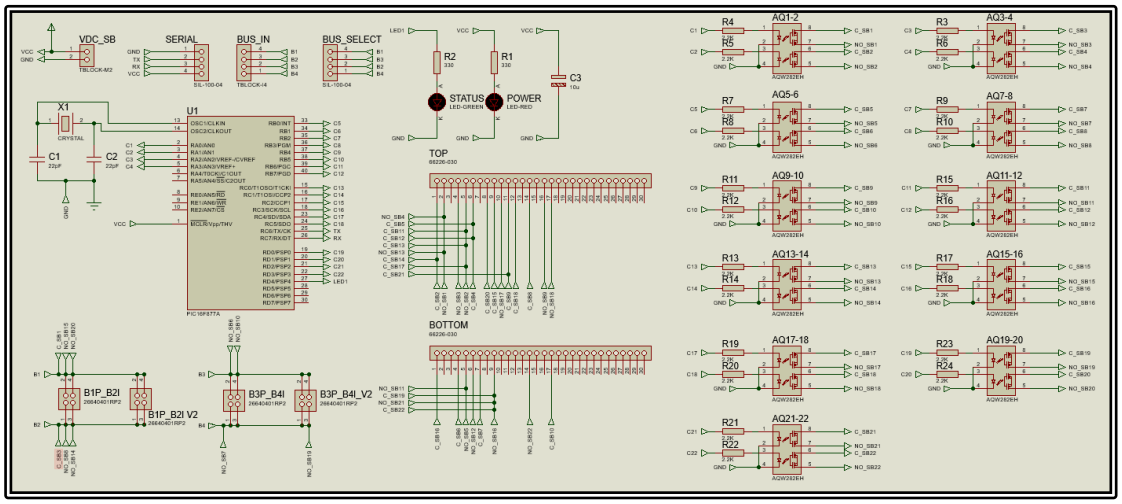
```
if (read_eeprom(0)=='s' && read_eeprom(24)=='e'){
   PORTA=control_salidas(2,4);
   PORTB=control_salidas(6,8);
   PORTC=control_salidas(14,6);
   PORTD=control_salidas(20,4);
}
```

Fuente: elaboración propia.

3.6. Construcción del prototipo

La etapa de construcción del prototipo inicia con el diseño esquemático sobre el software de simulación Proteus 8 Professional, en la figura 26 se observa el orden de las conexiones

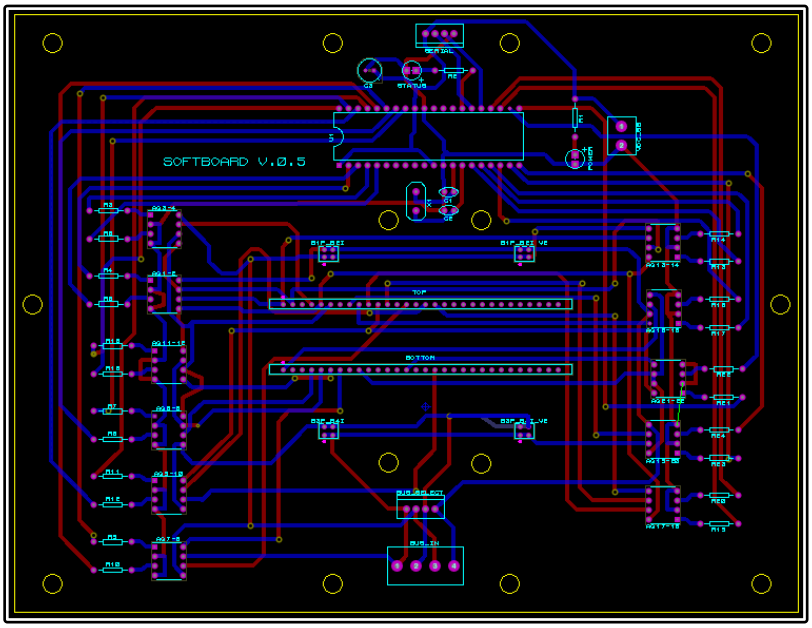
Figura 26: Esquemático para PCB



Fuente: elaboración propia

Dada la cantidad de pistas se diseñó una PCB impresa por ambas caras, en la figura 27 se presenta su diseño:

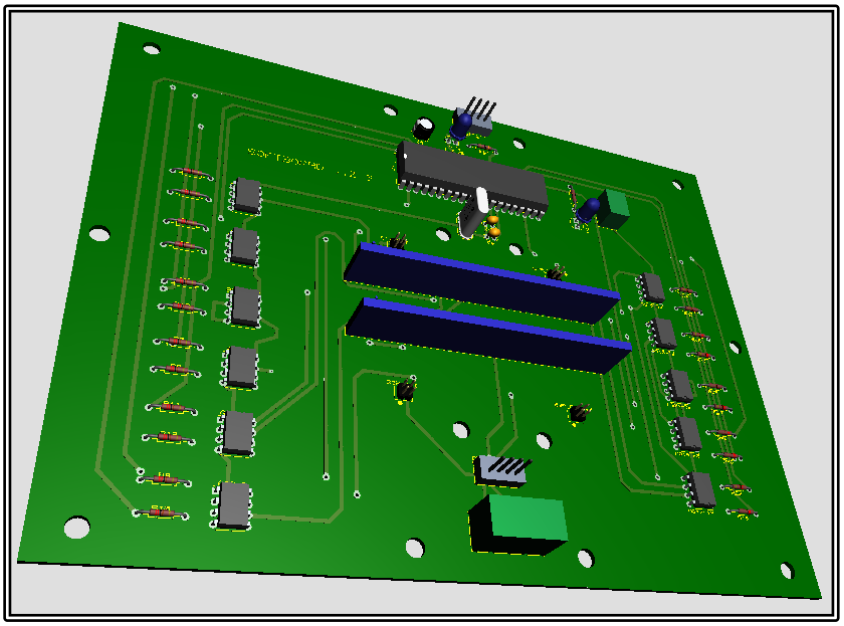
Figura 27. Diseño de PCB



Fuente: elaboración propia

El diseño 3D generado por el software se presenta en la figura 28.

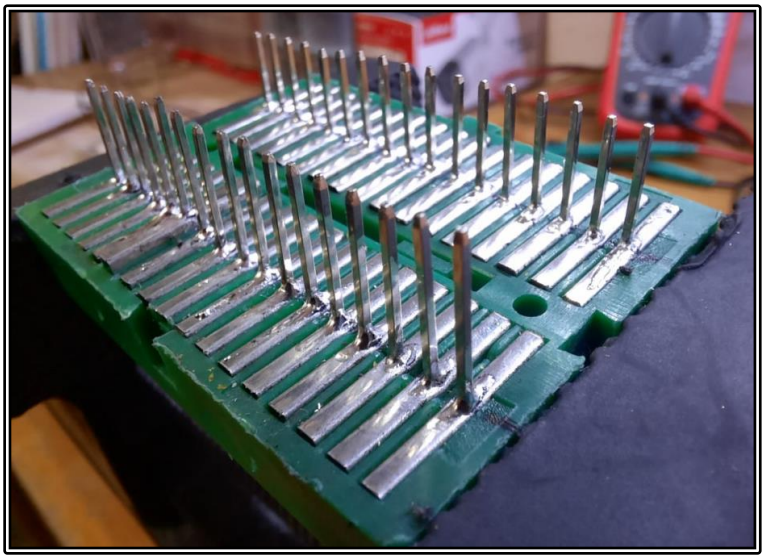
Figura 28: Visualización 3D



Fuente: Autor

La adaptación de la protoboard a la PCB se realizó incrustando y soldando pines bajo las láminas de conexión como se ve en la figura 29.

Figura 29: Pines soldados a las laminas



Fuente: elaboración propia

Finalmente, la puesta de componentes y verificación de funcionamiento tomo alrededor de 20 horas.

Figura 30: Construcción de la softboard

Fuente: elaboración propia

4. Pruebas, resultados y discusiones

4.1. Circuitos de prueba

Se realizó una selección de circuitos básicos fundamentales para realizar las pruebas en el simulador Cadence y el prototipo de Softboard, quedando seleccionados:

- Circuitos rectificadores con diodos
- Amplificador con BJT
- Amplificadores operaciones (Oscilador Wien)
- Oscilador 555
- Circuitos resistivos
- Circuitos RC

Sobre estos circuitos se tiene el anexo 1, donde podemos encontrar los resultados de sus pruebas. A continuación, se presentan los resultados de un regulador de voltaje a 12VDC, desde la etapa de rectificación.

4.1.1. Diseño

Se diseño una fuente de voltaje regulada de 12V con los siguientes parámetros:

Tabla 4. Parámetros de la fuente

Ítem		Valor
Voltaje máximo	Vp	20,65V
Voltaje eficaz de salida	Vrms	14,58V
Frecuencia	F	60Hz
Resistencia de carga	RL	10KΩ
Factor de rizado	Fr	10,65%

Fuente: elaboración propia

Cálculos:

$$Vout = Vp - 1.4V = 19.25V \qquad (1)$$

$$Vdc = 2 * \frac{Vp}{\pi} = 12.3Vdc \qquad (2)$$

$$Idc = \frac{Vp}{RL} = 1.23mA \quad (3)$$

$$Vr = \frac{Fr*Vout}{100\%} = \frac{10.65\%*19.25}{100\%} = 2.05V \quad (4)$$

$$C = \frac{Idc}{Vr} * F = \frac{1.23mA}{2.05V*60Hz} = 10uF \quad (5)$$

En la ecuación (1) se obtiene el voltaje de salida máxima del rectificador.

Con la ecuación (2) se obtiene el voltaje medio DC

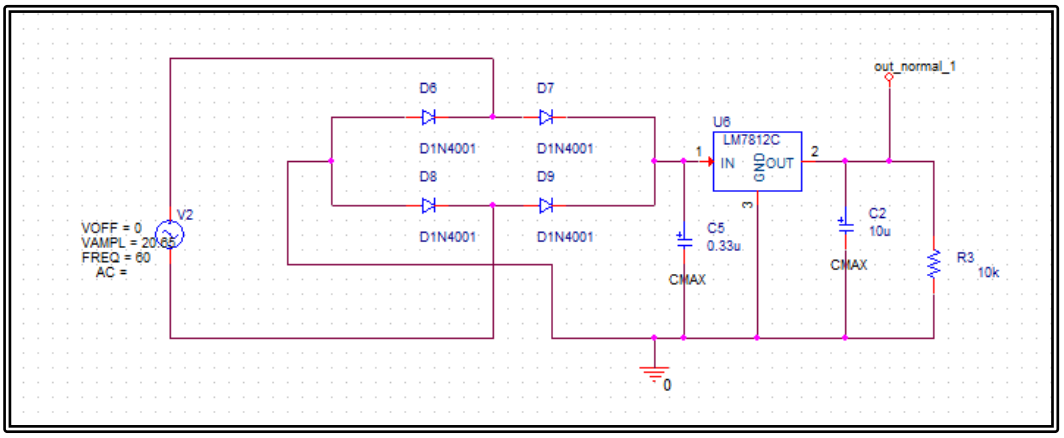
En la ecuación (3) se obtiene la corriente media DC

A partir de la ecuación (1), y los parámetros de la tabla, se obtiene el voltaje de rizado (4).

Para finalizar, el condensador de rizado (5) es obtenido a partir de (3) y (4) e información de la tabla de parámetros.

En la figura 31 se muestra el esquemático del circuito diseñado sin conexiones con SSR

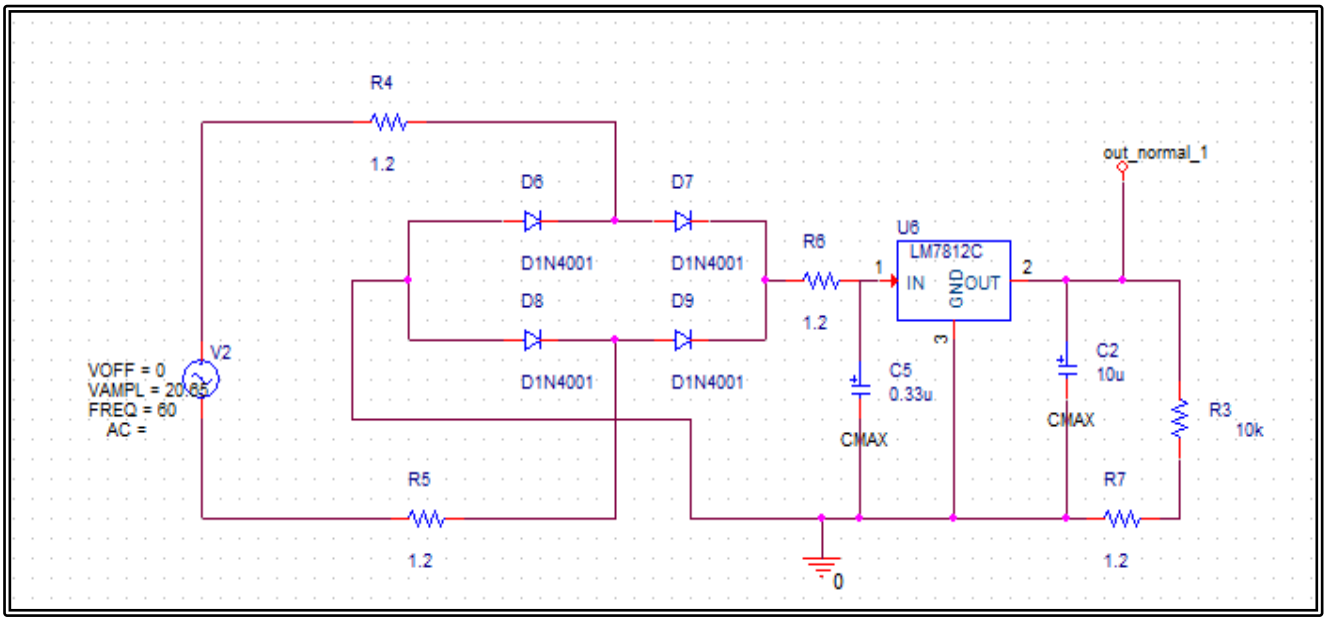
Figura 31. Esquemático circuito de prueba



Fuente: elaboración propia

En la figura 32 se agrega una resistencia de 1.2Ω en cada conexión que se realiza por medio de SSR. En total para este circuito se realizaron 4 conexiones.

Figura 32. Esquemático circuito de prueba con resistencia simulando el SSR

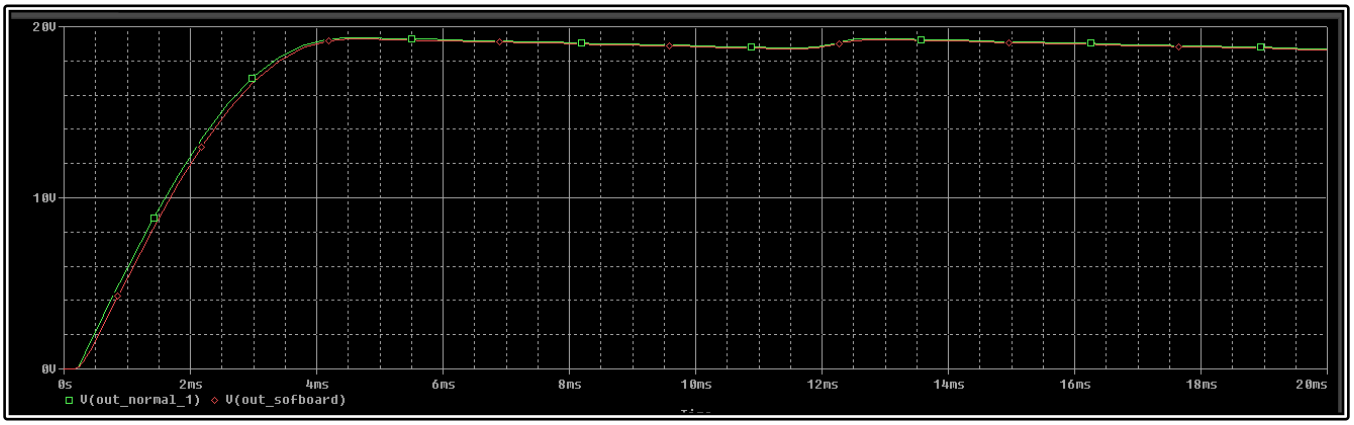


Fuente: elaboración propia

4.1.2. Simulación

La salida rectificada se presenta en la figura 34:

Figura 34. Voltaje de salida sin regular



Fuente: elaboración propia

En la figura anterior se tiene la señal de salida del circuito normal en verde y en rojo la salida en la softboard. Se puede apreciar una pequeña diferencia entre la señal de la softboard y la señal normal ya que como se ve en la figura 32 se agregan 4 resistencias de 1.2Ω al circuito como parte del modelo del SSR.

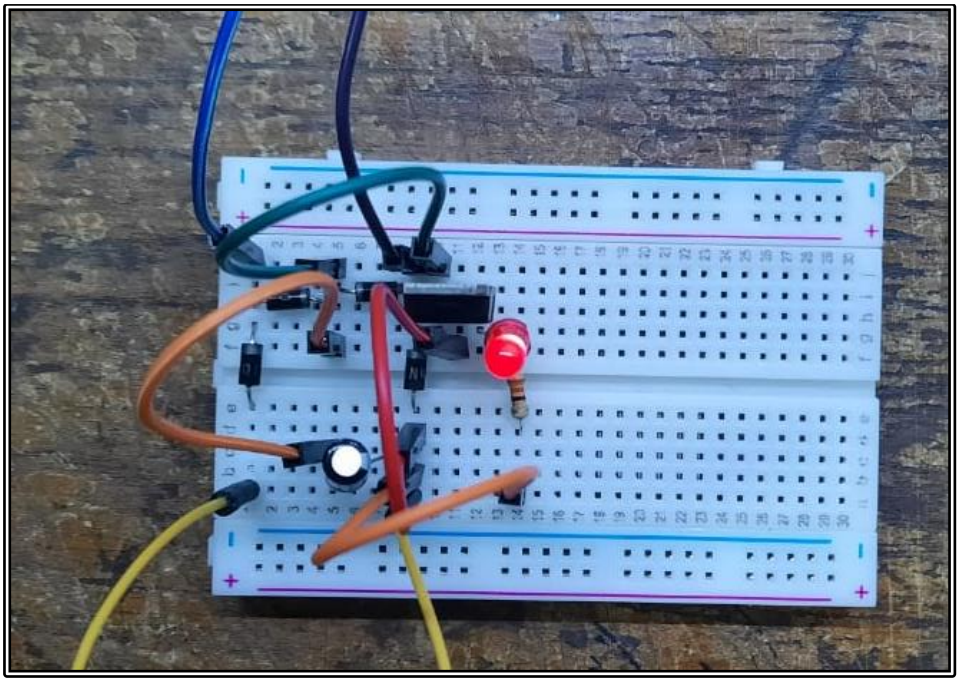
4.1.3. Montaje en protoboard

Para el montaje en protoboard se usaron los siguientes materiales:

- 4 diodos D1N4001
- 1 resistencia de 10K Ω
- 1 regulador LM7812
- Protoboard
- 8 conectores macho-macho
- Transformador entrada 110VAC y salida 14.6VAC

En la figura 35 se muestra el montaje realizado en la protoboard, con un tiempo de montaje promedio de 8 minutos.

Figura 35. Montaje de circuito de prueba en protoboard



Fuente: Elaboración propia

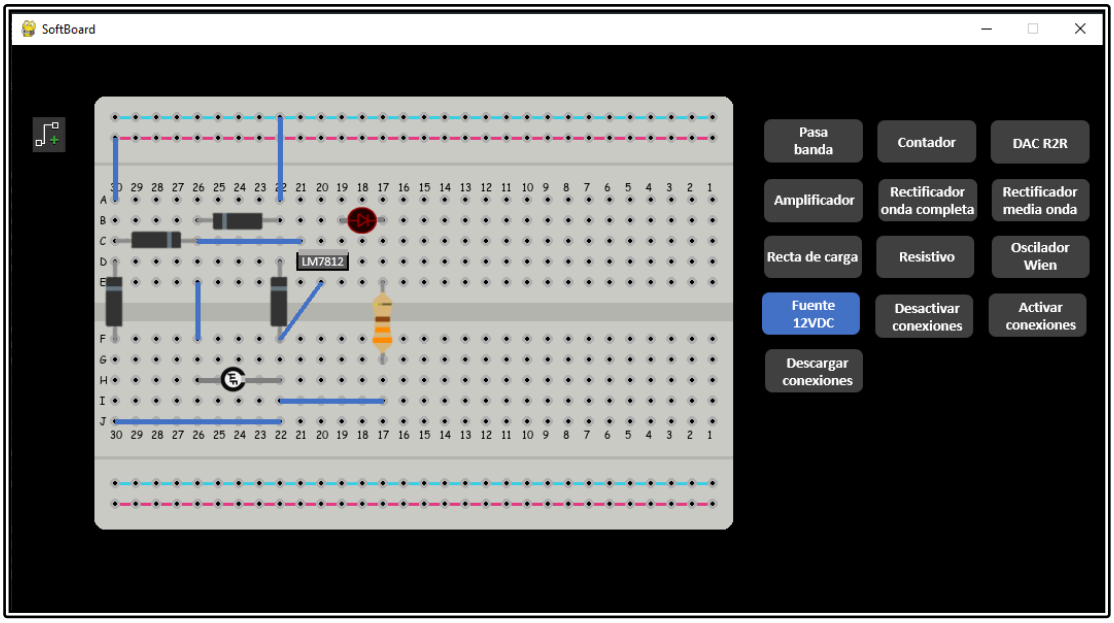
4.1.4. Montaje en Softboard

Para el montaje en softboard se usaron los siguientes materiales:

- 4 diodos D1N4001
- 1 resistencia de 10 K
- 1 regulador lm7812c
- Softboard
- Computador
- Transformador entrada 110Vac; salida 14.6Vac

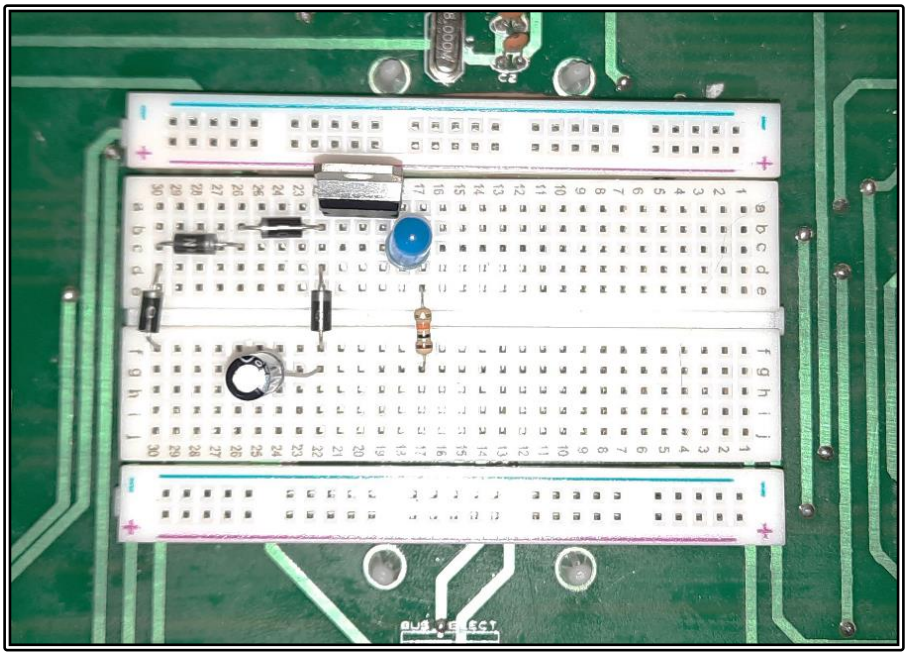
En la figura 36 se muestra el montaje cargado desde la interfaz a la softboard y en la figura 37 el circuito montado físicamente, el cual tuvo un tiempo de montaje promedio de 2 minutos y fue necesario utilizar un computador para cargar el circuito en el prototipo.

Figura 36. Circuito de prueba visualizado desde la GUI



Fuente: elaboración propia

Figura 37. Circuito de prueba montado en softboard



Fuente: elaboración propia

Para el montaje del circuito se siguen los pasos:

I. Conectar el cable serial-USB al computador y la softboard

II. Cargar el circuito correspondiente desde la interfaz de usuario

III. Verificar continuidad entre conexiones

IV. Desconectar el cable serial.

V. Realizar el montaje de los componentes en la posición que se indica en la interfaz de usuario.

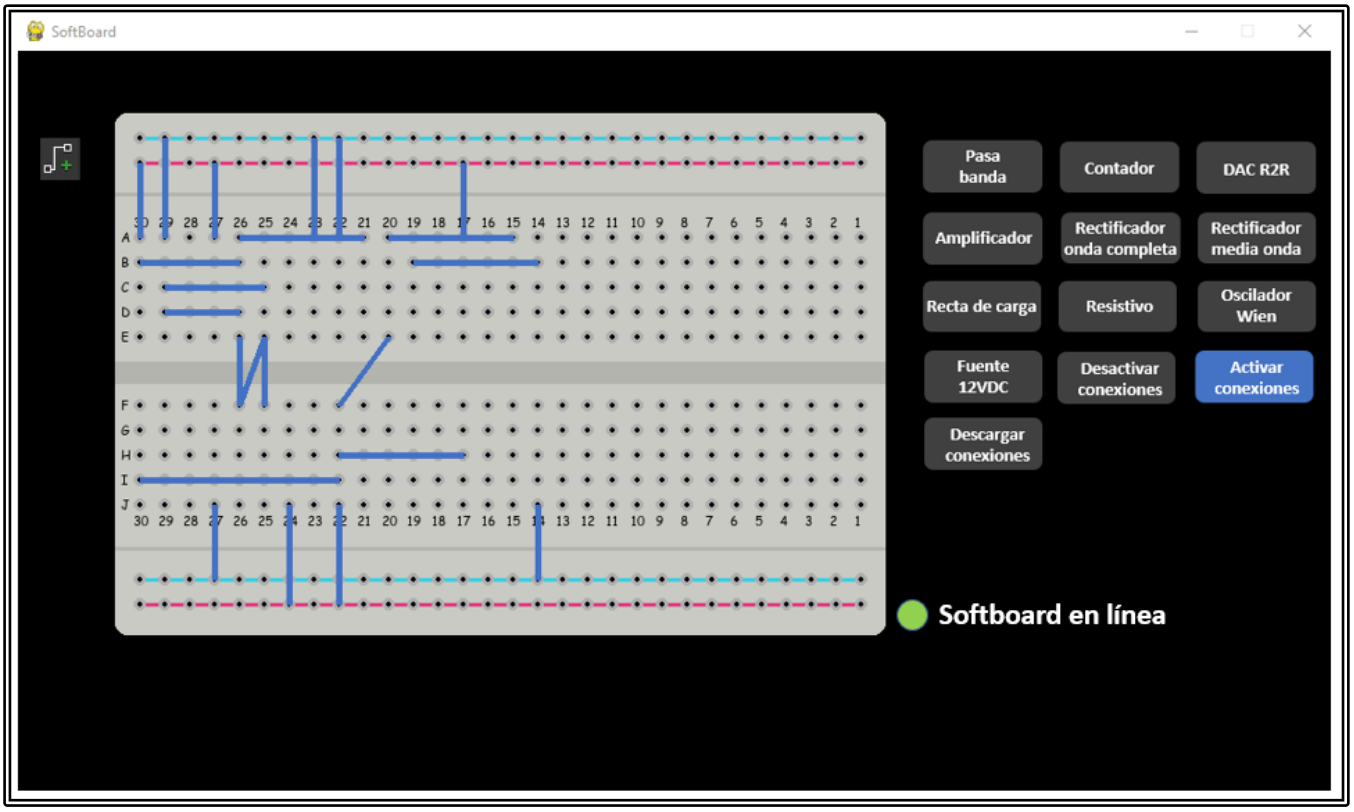
VI. Energizar el circuito conectando al bus de voltaje la salida del transformador.

4.2. Resultados

4.2.1. Interfaz gráfica de usuario

Se diseño una interfaz gráfica intuitiva y fácil de usar, como se ve en la figura 38:

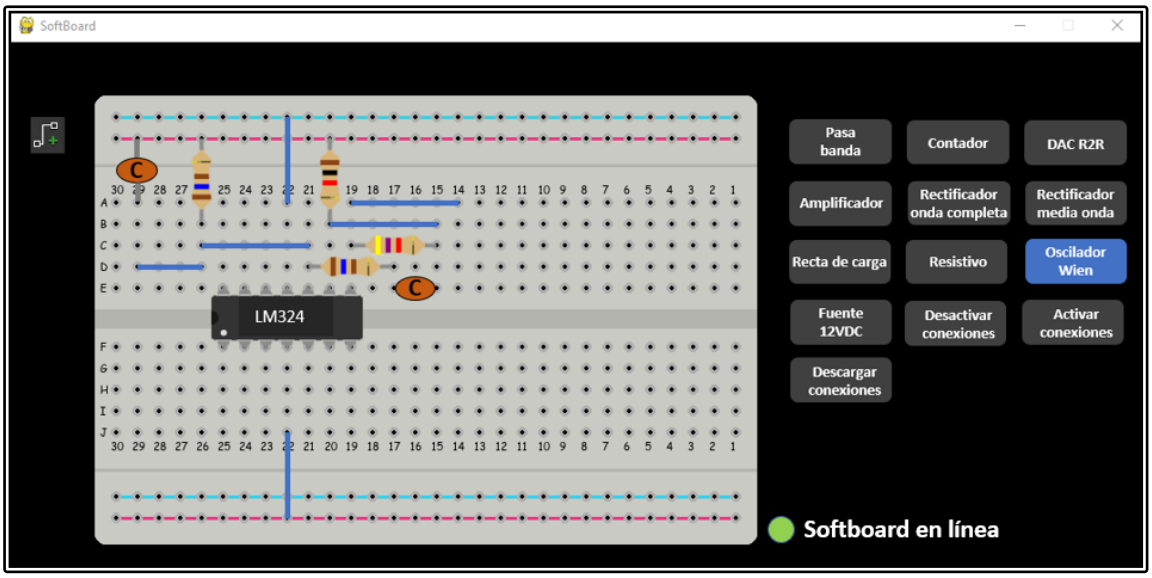
Figura 38. Interfaz gráfica con todas las interconexiones habilitadas



Fuente: elaboración propia

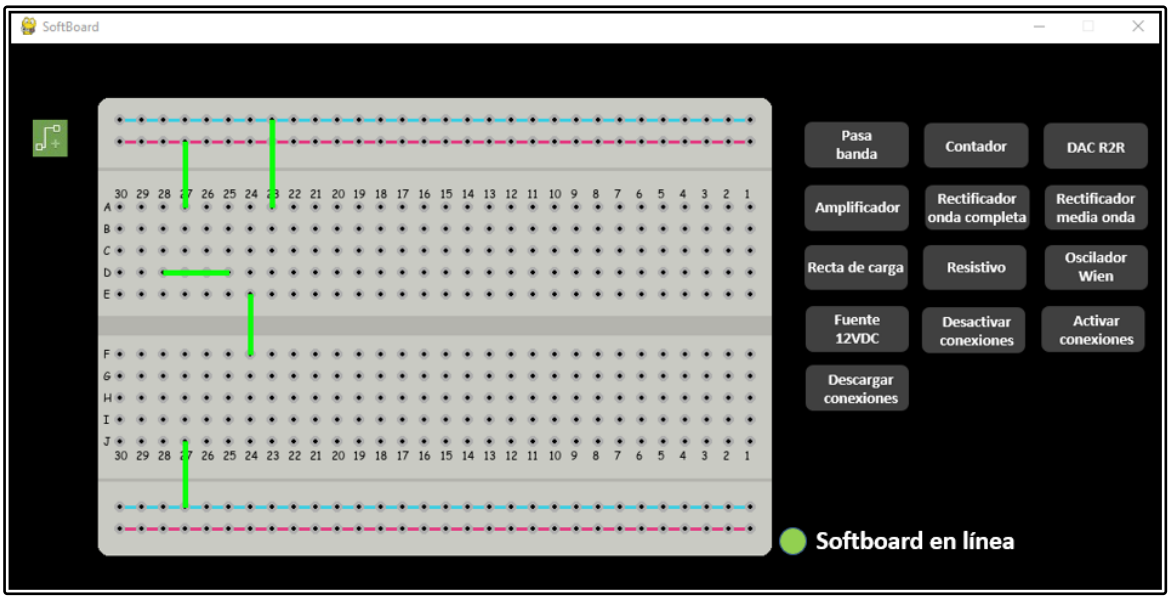
Al costado derecho se observa que está habilitado el botón “Activar conexiones” la interfaz gráfica dibuja las conexiones en color azul necesarias para la interconexión de los elementos, en simultaneo se envía el vector de configuración a microcontrolador encargado de energizar los SSR. En la misma sección se puede apreciar la opción “Descargar conexiones” esta descarga de la memoria EPROM del microcontrolador la información del ultimo vector de interconexiones enviadas. Allí también se tienen 10 circuitos sobre los cuales se realizan pruebas de funcionamiento, al pulsar sobre alguno se envía al microcontrolador el vector de las interconexiones para activar, estas se visualizan en la interfaz, acompañada de los componentes y su ubicación, como muestra la figura 39:

Figura 39. Circuito “Oscilador Wien” cargado en la GUI



Al costado izquierdo se tiene el botón “Cable”, este habilita el puntero del mouse para dibujar un cable en color verde en la interfaz, antes de ser enviada al microcontrolador pasa por un proceso de verificación, donde se determina si la interconexión está dentro de las 22 creadas para este modelo de la softboard, en la figura 40, se observa un ejemplo de uso del botón “Cable”.

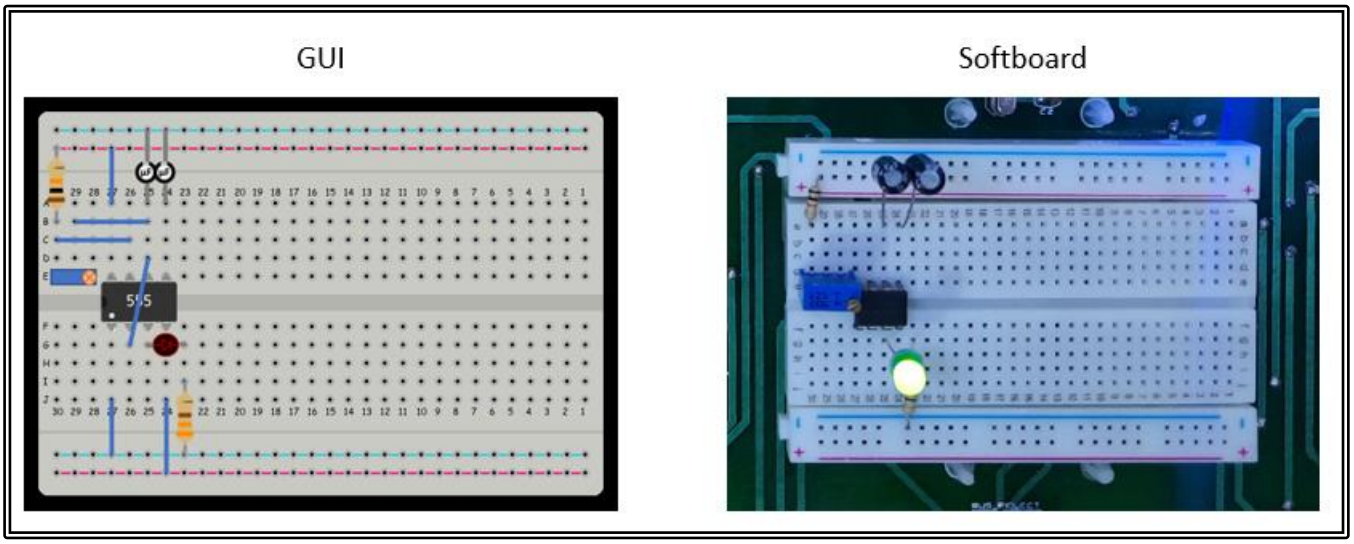
Figura 40. Ejemplo de funcionamiento botón “Cable”



Fuente: elaboración propia

Un montaje relacionando la visualización del software con la softboard se ve en la figura 41:

Figura 41. Visualización GUI y softboard



Fuente: elaboración propia

4.2.2. Algoritmo del microcontrolador para control

El código fuente de la figura 25 asigna un valor decimal al puerto, dependiendo cual relé se quiere activar. El valor decimal se obtiene multiplicando el valor del carácter, ‘0’ o ‘1’, por la potencia de 2 correspondiente, por ejemplo:

Datos en la EEPROM→['s', '0','1','0','1','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','e']
*Salida para el puerto A→ 0*2^0+1*2^1+0*2^2+1*2^3 = 5 →Valor en binario= 0101*

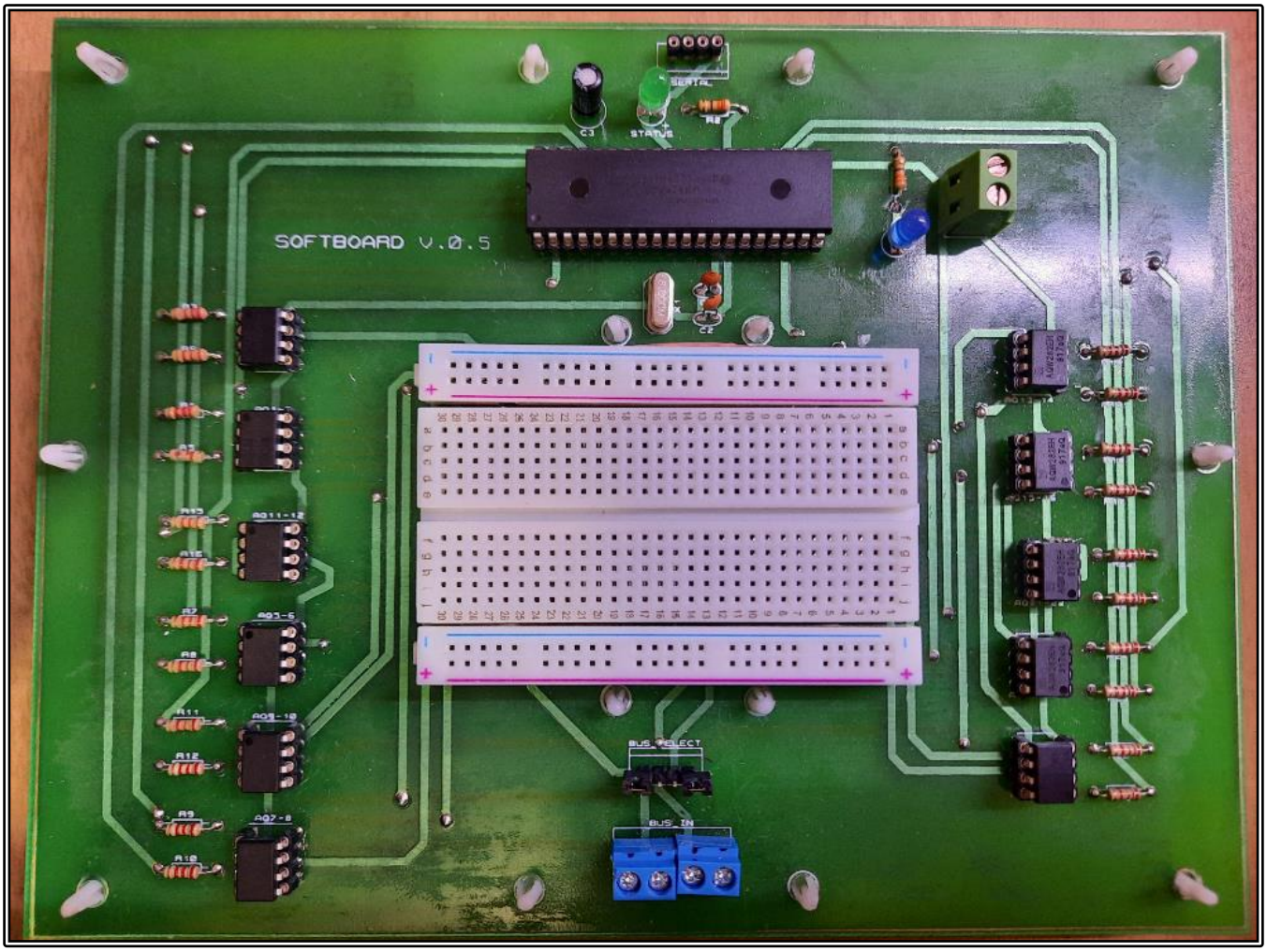
Al asignarle el valor '5' al puerto A se activan las salidas correspondientes al pin A1 y A3 y se desactivan las salidas A0 y A2.

El algoritmo realizado cumple satisfactoriamente con la necesidad del proyecto.

4.2.3. Construcción del prototipo

La aplicación de la metodología propuesta dio como resultado el prototipo de la Softboard, que se puede apreciar en la figura 42:

Figura 42. Softboard construida



Fuente: elaboración propia

La softboard está compuesta por:

1) ***Microcontrolador****:* un PIC16F877A

2) ***Interfaz de comunicación con el software y led de estado:*** consiste en una comunicación RS-232 a USB, con led de estado que parpadea una vez en un segundo para indicar que envía datos al software y dos veces en un segundo para indicar que recibe datos.

3) ***Relés de estado sólido (SSR):*** Es un circuito integrado Panasonic AQW282EH.

4) ***Alimentación softboard y led de estado:*** Para energizar la softboard se requiere de 5VDC a 100mA. El led de estado enciende cuando esta energizada y se apaga cuando no lo está.

5) ***Bus de voltajes:*** las borneras de bus de voltaje conectadas con las filas superiores (bus **a**) e inferiores (bus **b**) de la protoboard como se indica en la figura 44, son configurables en tres modos:

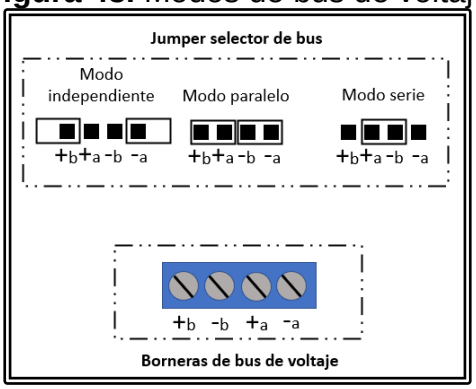
Independiente: Cada bus se conecta a las filas positiva y negativa de la parte inferior y superior.

Paralelo: Las filas positivas y negativas están conectadas entre sí.

Serie: Las filas positiva superior y negativa inferior están conectadas entre sí.

Para explicar mejor la configuración de los modos se tiene la figura 43.

Figura 43. Modos de bus de voltaje



Fuente: elaboración propia

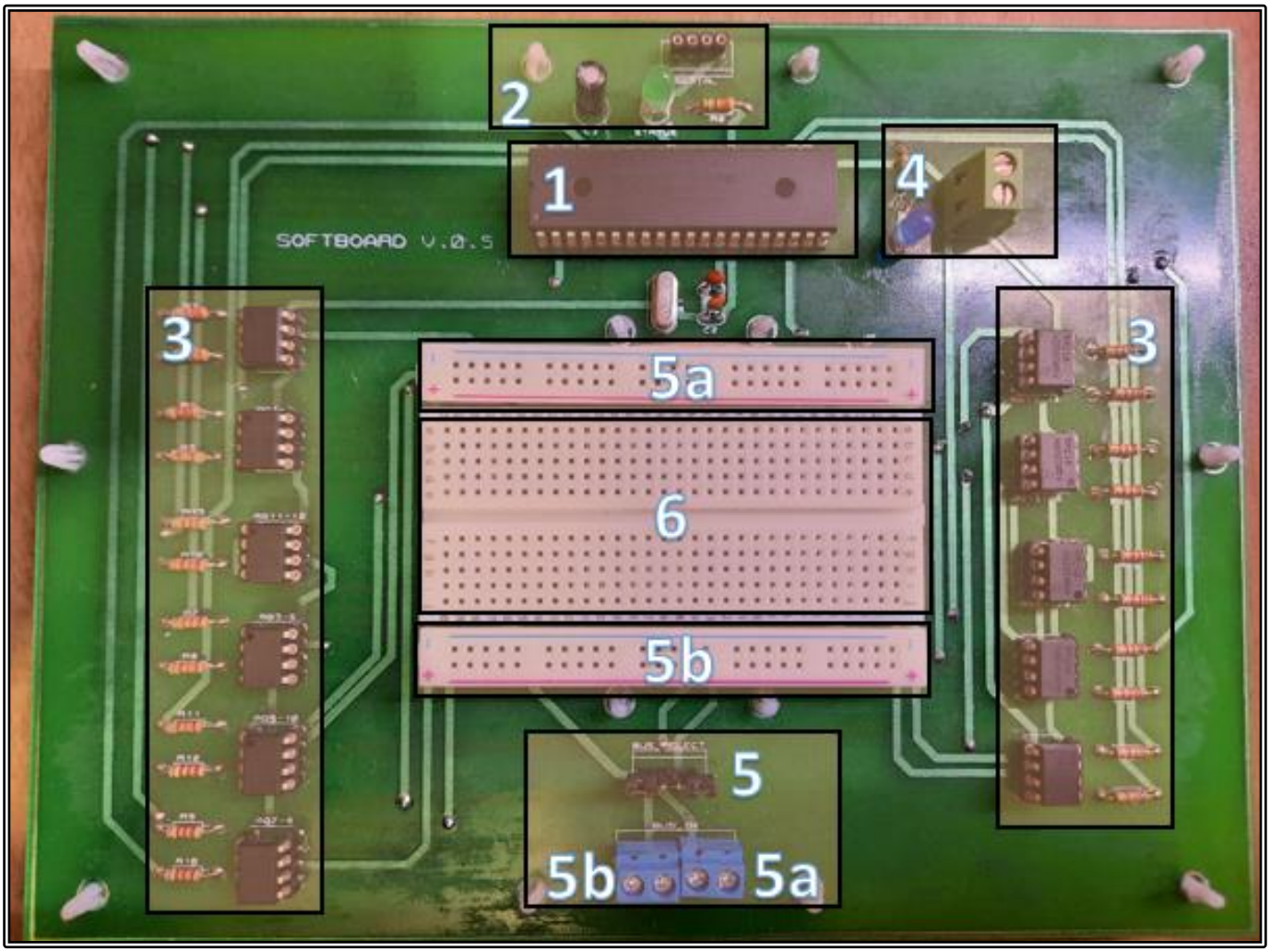
5a. Bus de voltaje a: Conectado el voltaje en las borneras se energiza el bus superior de la protoboard.

5b. Bus de voltaje b: Conectado el voltaje en las borneras se energiza el bus inferior de la protoboard.

6) Protoboard: Espacio de ubicación para los componentes.

Se puede apreciar las partes nombradas anteriormente en la figura 44:

Figura 44. Partes de la softboard

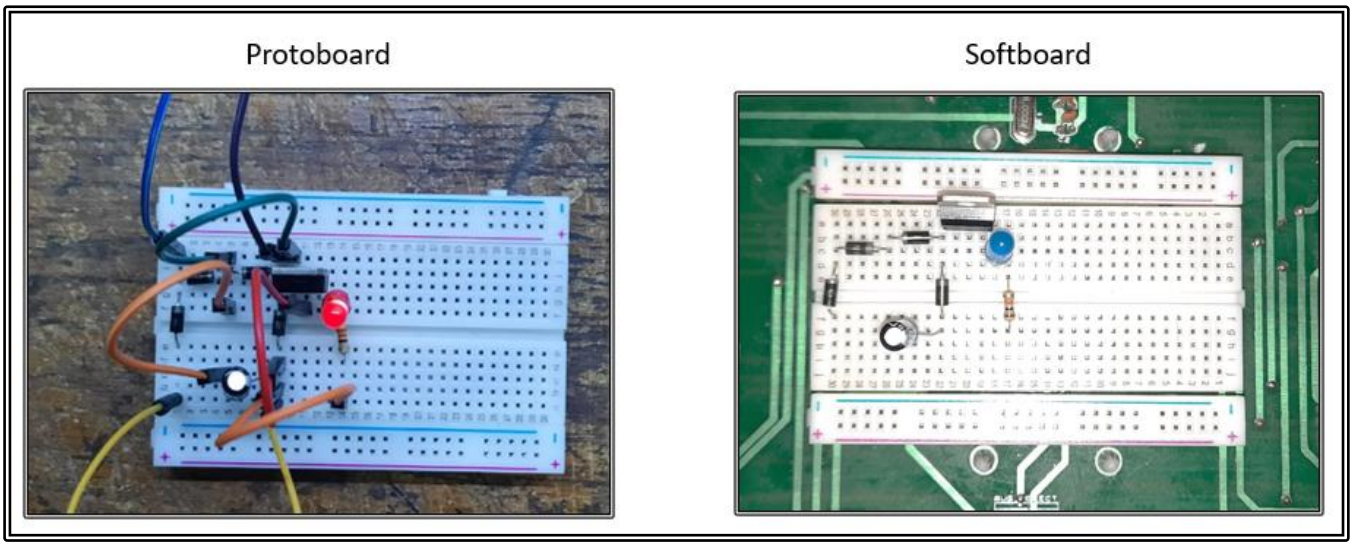


Fuente: elaboración propia

4.2.4. Comparación entre softboard y protoboard

La primera comparación que realizamos es la física, evidenciando el cambio de espacio de trabajo y aspecto visual como se observa en la figura 45, con el montaje del circuito de prueba.

Figura 45. Comparación protoboard y softboard con el circuito de prueba.

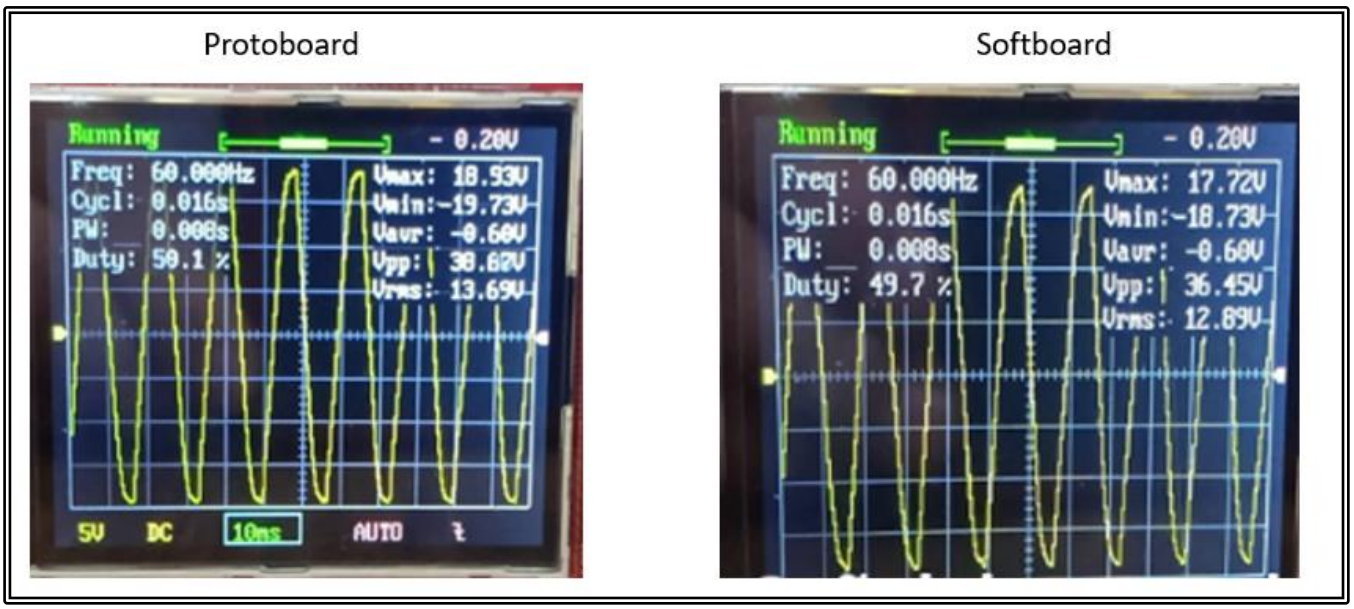


Fuente: elaboración propia

Para realizar el montaje de este circuito es necesario el uso de 8 cables conector en la protoboard, para el caso de la softboard se usó el modo paralelo de bus de voltaje y 6 conexiones de SSR.

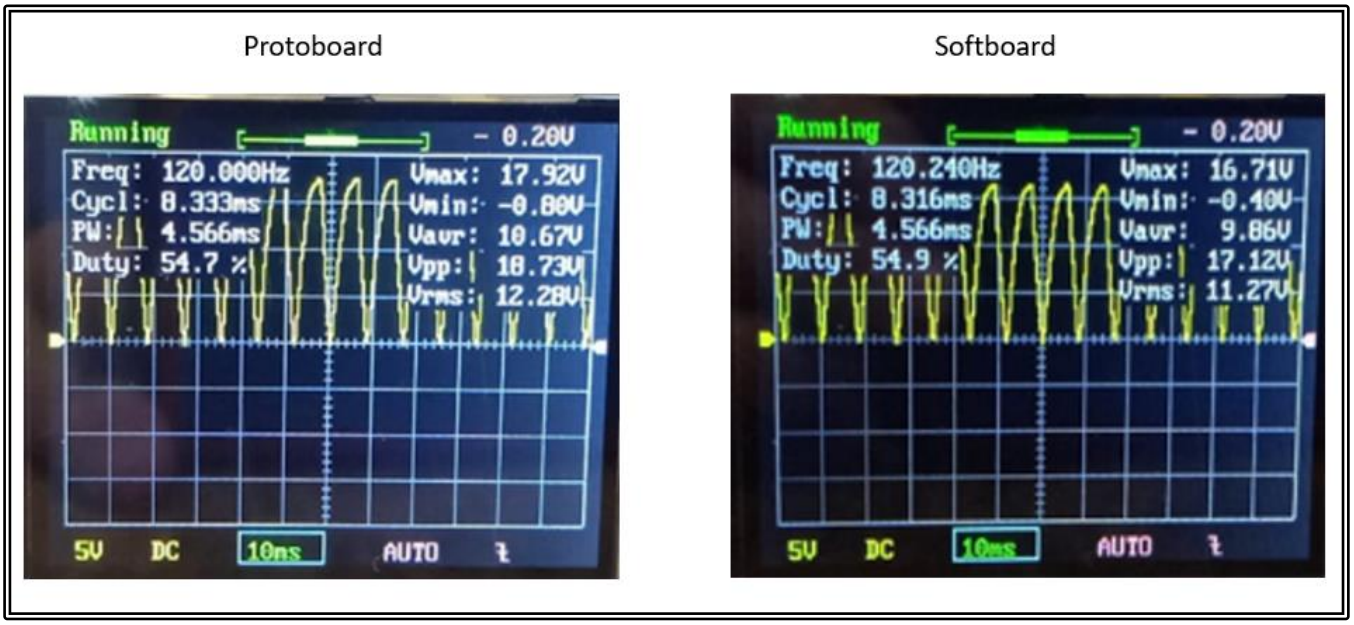
Luego se procede a realizar comparación de cada montaje con las mediciones realizadas con osciloscopio, de tal modo se tiene:

Figura 46. Comparación de señal de entrada al puente rectificador.



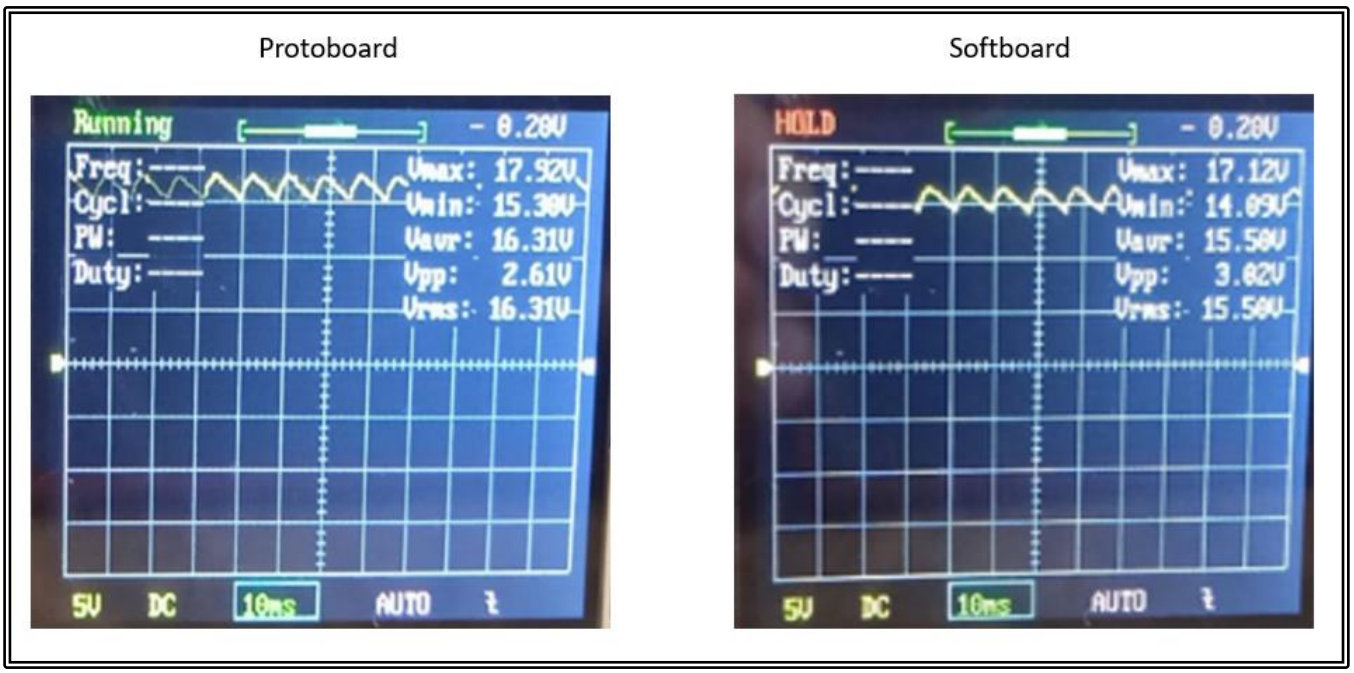
Fuente: elaboración propia

Figura 47. Comparación señal de salida del puente rectificador.



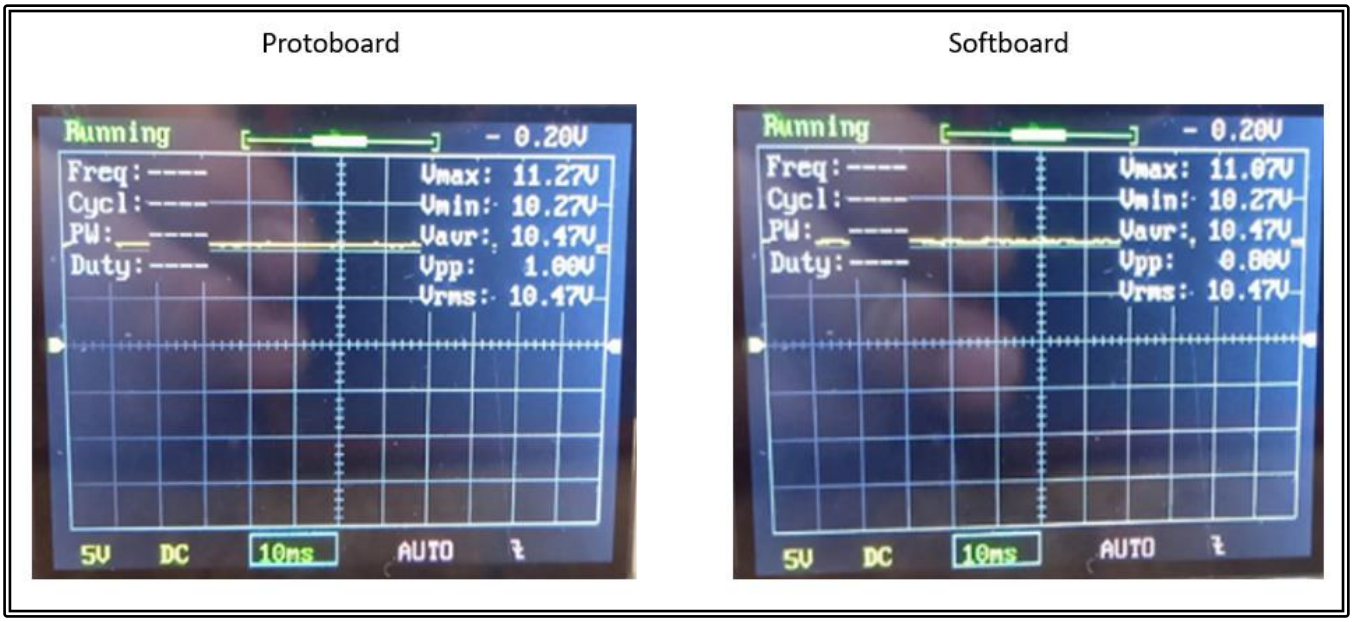
Fuente: elaboración propia

Figura 48. Comparación señal de entrada al regulador.



Fuente: elaboración propia

Figura 49. Comparación señal de salida del regulador



Fuente: elaboración propia

La tabla 5 muestra el error porcentual de los valores medidos en la protoboard y los medidos en la softboard. Para este cálculo se tomó el valor de la protoboard como valor real y el valor medido en la softboard se tomó como valor practico, además los montajes se realizaron con los mismos componentes en ambos casos para acercarse más al error porcentual real.

$$\%error = \frac{|\text{valor protoboard} - \text{valor softboard}|}{\text{valor protoboard}} * 100$$

Tabla 5. Errores porcentuales

Errores porcentuales				
Ítem	Medición protoboard	Medición softboard	% error	Unidad
Voltaje pico de entrada	18,01	17,72	1,64%	VAC
Vrms	13,2	12,89	2,40%	VAC
Voltaje máximo rectificado	17,32	16,71	3,65%	VAC
Voltaje mnáximo filtrado	17,42	17,12	1,75%	VAC
Voltaje mínimo filtrado	15,45	15,09	2,39%	VAC
Voltaje de rizado	1,97	2,03	-2,96%	VAC
Voltaje regulado	11,27	11,07	1,81%	VDC
Frecuencia	60	60	0,00%	Hz
Corriente salida regulador	945	921	2,61%	uA DC
Corriente total	4,38	4,26	2,82%	mA AC

Fuente: elaboración propia

En la tabla anterior se evidencia un error porcentual bajo para las distintas medidas. Estas diferencias se deben a la resistencia de salida del SSR y los instrumentos de medición.

4.3. Discusiones

4.3.1. Comodidad

- Durante los montajes de circuitos de pruebas se siente más comodidad al usar las herramientas de medición (multímetro, osciloscopio, entre otros), esto porque no hay cables.
- Hay una sensación de incertidumbre al querer ver si las interconexiones están bien realizadas, porque no se ve un cable entre los componentes.

4.3.1. Tiempo

- El tiempo de montaje se reduce drásticamente, pues solo hay que ubicar componentes
- Parte del tiempo ahorrado se debe invertir en la organización de componentes en el software.

4.3.1. Limitaciones

- Se crea una dependencia del software y por lo tanto de un computador.
- Se pierde la interacción con herramientas como pelacables, alicates o cortafríos.
- El prototipo actual está limitado a una serie de circuitos prediseñados y 22 conexiones disponibles
- Los circuitos a montar en la softboard no deben superar los 400mA.

4.3.1. Costos

- Los costos de la construcción de este prototipo se relacionan en la tabla 6:

Tabla 6. Costos del prototipo

Item	Descripción	Cantidad	Precio UN	Precio T UN
1	Microcontrolador PIC16F877A	1	$ 12.000	$ 12.000
2	Max 232	1	$ 12.000	$ 12.000
3	Cable USB-Serial	1	$ 18.000	$ 18.000
4	Protoboard	1	$ 8.000	$ 8.000
5	PCB	1	$ 50.000	$ 50.000
6	Componentes varios (resistencias, pines, borneras, postes, condensadores, cristal, leds)	1	$ 35.000	$ 35.000
7	Rele de estado sólido AQW282EH	20	$ 12.004	$ 240.080
8	Gastos de importación (desde EEUU)	1	$ 174.870	$ 174.870
9	Horas de trabajo estudiante	200	$ 9.145	$ 1.829.000
10	Horas de trabajo docente	10	$ 18.000	$ 180.000
11	Insumos	1	$ 40.000	$ 40.000
12	Alquiler herramienta y equipos de computo	1	$ 300.000	$ 300.000
13	Viaticos	1	$ 150.000	$ 150.000
			TOTAL	$ 3.048.950

Fuente: elaboración propia

- El 72.8% del costo de los SSR (ítem 7 y 8) se gasta en la importación, esto es una gran desventaja por nuestro país, incluso, para el resto de componentes en el mercado internacional se puede tener un ahorro.
- El 70.8% del costo total es invertido en mano de obra (ítem 9, 10 y 13), teniendo en cuenta que esta es una etapa de desarrollo.
- El costo total del proyecto es alto, si se trata de hacer una comparación del costo de adquisición de una softboard, sin embargo, se debe tener en cuenta que se puede hacer una mayor investigación de mercado e ingeniería, para obtener costos mediante producción masiva o incluso el desarrollo de una ASIC (Circuitos Integrados de Aplicación Específica).

5. Conclusiones

Este trabajo ha desarrollado una herramienta que toma ventaja en practicidad, velocidad, comodidad y seguridad para el montaje de circuitos electrónicos respecto a una protoboard, es practica ya que posee una interfaz gráfica que apoya la implementación de circuitos de aspecto visual similar a la realidad, la “softboard”

intenta evolucionar el método bajo el cual se realizan pruebas de laboratorio. Es necesario hacer investigaciones de diferentes circuitos más a fondo, ya que por la pandemia no se contó con herramientas ideales como generador de ondas y osciloscopio de calidad.

La Softboard puede llegar a convertirse en una nueva herramienta para el prototipado electrónico, pero requiere más investigación y es necesario utilizar otras tecnologías que permitan reducir los costos y las limitaciones. Con la tecnología implementada en el prototipo se obtuvieron buenos resultados, con porcentajes de error bajos como se muestra en la tabla

5, lo que impulsa a seguir con la investigación y así desarrollar un producto que pueda cumplir con las necesidades del mercado.

5.1. Trabajos futuros

- Integrar el software desarrollado con otro como el “Automatic Protoboard Layout from Circuit Schematics” [3].
- Integrar diseño de circuitos con archivos basados en PSpice.
- Diseñar un ASIC.
- Investigar mercado para estudio de viabilidad del producto.
- Conseguir financiación publicando el proyecto en plataformas de crownfoundig o micromecenazgo.

Referencias

[1] W. Young, "¿What is a 'Breadboard'?," *¿What is a "Breadboard"?*, 2011. https://tangentsoft.net/elec/breadboard.html.

[2] A. Robbins and W. Miller, "ANÁLISIS DE CIRCUITOS Teoría y práctica," in *ANÁLISIS DE CIRCUITOS Teoría y práctica*, Cuarta., Cengage learning, 2007, p. 16.

[3] M. Mekonnen, "Automatic Protoboard Layout from Circuit Schematics," no. 2013, 2014.

[4] A. Muñoz Muñoz and M. Díaz Perea, "Metodología por proyectos en el área de conocimiento del medio," *Rev. docencia e Investig.*, vol. 19, pp. 101–126, 2009.

[5] M. A. Angarita Velandia, F. H. Fernández Morales, and J. E. Duarte, "Formación de ingenieros interdisciplinarios a través de una metodología activa con temáticas integradoras," *Saber, Cienc. y Lib.*, vol. 11, no. 2, pp. 177–187, 2016, doi: 10.18041/2382-3240/saber.2016v11n2.555.

[6] S. Leon, M. Enriquez, J. Monagas, and A. Vega, "SISTEMA DE PROTOTIPADO ELECTRÓNICO PROTO3D®; APLICACIONES A LA DOCENCIA," Universidad de Las Palmas de Gran Canaria, 2014.

[7] E. Blauvelt, "(12) United States Patent," US 6,685,483 B2, 2004.

[8] "PyGame." https://www.pygame.org.