



UNIVERSIDAD NACIONAL DE CÓRDOBA
FACULTAD DE CIENCIAS EXACTAS, FÍSICAS Y NATURALES
CÁTEDRA DE ELECTRÓNICA DIGITAL I

TRABAJO PRÁCTICO N° 2

**“CIRCUITOS SECUENCIALES - MAQUINAS DE
ESTADOS - ALARMA”**

Grupo N° 3

Alumnos:

Langleib, Emilia

Rojo, Juan Pablo

Saillen, Simón

Vera, Violeta

NOTA:

Profesor:

Ing. Vrech, Ruben

Comisión: Viernes 9:30 hs

Abril / 2022

Consigna

El objetivo de este Trabajo Práctico es diseñar y construir en plaqueta tipo “Protoboard” un circuito secuencial como autómata de Moore tipo Alarma.

Dicha alarma deberá constar de un teclado matricial con código de activación/desactivación, 3 LED que describan el estado en que se encuentra la alarma, un sensor de disparo y un botón de pánico.

El circuito deberá ser realizado bajo el método de Flip-Flop JK, se deberá utilizar una sola variable externa y la señal de Clock deberá ser generada por el teclado matricial.

Desarrollo

1. Principios de funcionamiento de la Alarma

Teclado Matricial:

Para el funcionamiento del teclado matricial 4x4, decidimos utilizar un codificador a 4 bits. Este recibe información sobre las columnas y filas del teclado matricial, y la traduce a código binario de 4 bits.

Dicho codificador también incluye un eliminador de rebote mediante 2 capacitores conectados a tierra desde KBM y OSC, por *datasheet* KBM (KeyBounce Mask) debe ser 10 veces mayor que el conectado en OSC (Oscillator), por conveniencia y debido a que ya poseíamos un capacitor de $10\mu\text{F}$ utilizamos ese en KBM y conectamos uno de $1\mu\text{F}$ en OSC. También utilizaremos una salida del codificador llamada DA (Data Available) que nos indicará cada vez que es presionada una tecla, generando así la señal de Clock que será conectada a los Flip-Flops JK.

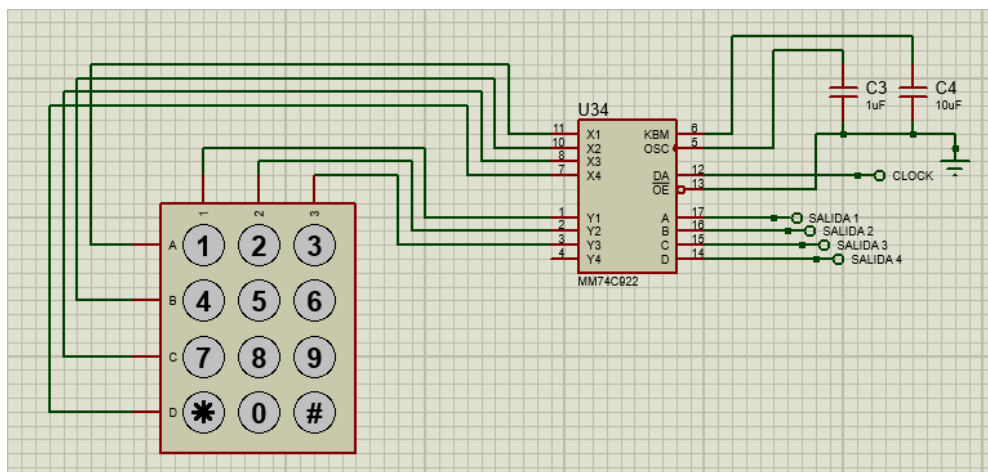


Imagen 1: Teclado Matricial y Codificador

Comparadores:

Debido a la necesidad de diferenciar 2 valores del resto en el teclado matricial, optamos por el uso de 2 comparadores binarios de 4 bits, el funcionamiento de estos es de tal manera que a los datos ingresados desde el teclado (pasados por el codificador), se los compara con un valor binario modelo elegido (en nuestro caso solo nos importa el valor de “0” y de “3”), si el valor ingresado por el teclado es igual al modelo entonces se devolverá “1” en la salida, en caso contrario, se devolverá “0”. Cabe aclarar que si bien se permiten realizar otras comparaciones en el mismo chip, sólo utilizaremos las igualaciones.

También aclaramos que cada valor modelo es asignado por el codificador, y se encuentra en la *datasheet* del mismo.

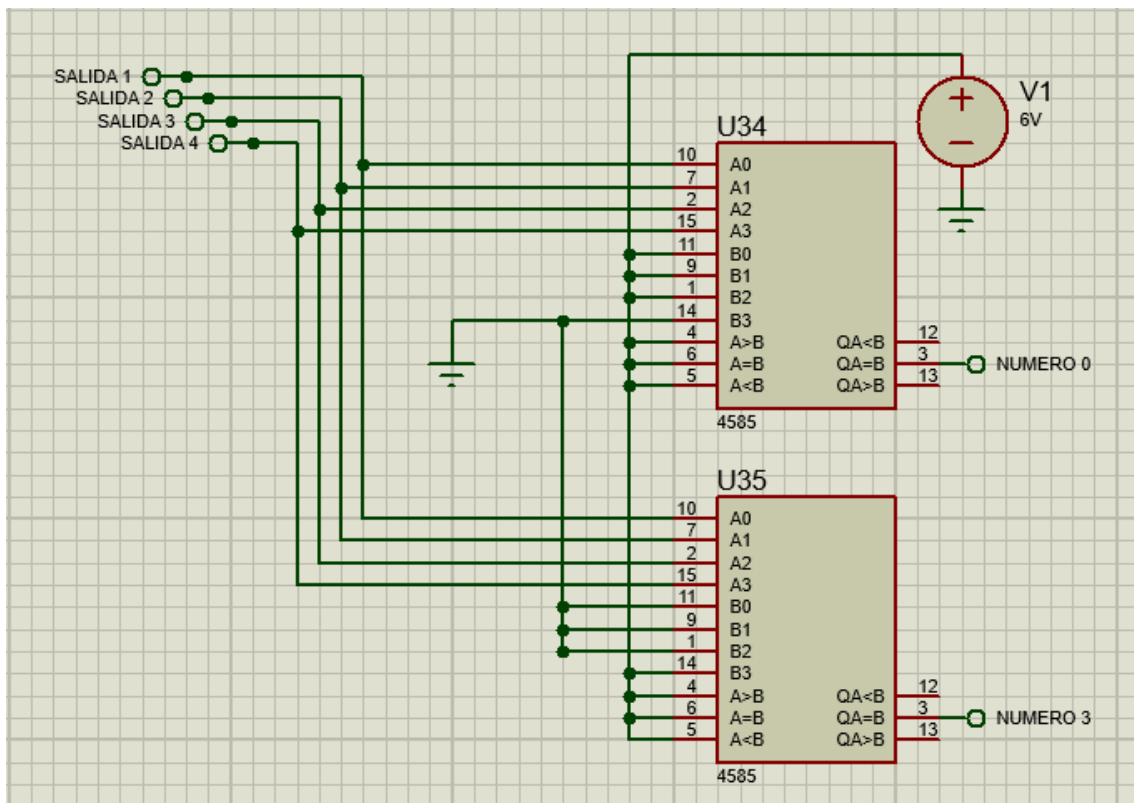


Imagen 2: “Comparadores de 4 bits”

Si bien en la simulación los valores binarios modelos para los números “0” y “3” son 1110 y 0001 respectivamente, en el circuito físico dichos valores son 1011 y 0100 (por *Datasheet*). Por lo que las conexiones mostradas en la *Imagen 2* no son las realizadas en el circuito físico.

Variable Externa:

Partiendo de la consigna que solo se puede utilizar una sola variable externa, nos presentamos ante la complicación de diferenciar cuándo esta tiene valor 1 y valor 0, sobretodo teniendo en cuenta que valdrá 1 cada vez que se presione el número “0” o “3” requeridos para la activación o desactivación de la alarma, y deberá valer 0 en caso de que se presione alguna otra tecla o se presione en orden equivocado.

Para resolver dicho problema, diseñamos un circuito combinacional que compara el valor de Q_1 presente en el FF1 con la tecla necesaria en cada estado determinado para pasar al siguiente.

Por ejemplo, si nos encontramos en el estado $E_0 = 000$, cuyo valor de Q_1 es 0, la variable X sólo valdrá 1 si se presiona la tecla “0”, y si nos encontramos en el estado $E_2 = 010$, cuyo valor de Q_1 es 1, necesitaremos presionar la tecla “3” para pasar al siguiente estado, en caso contrario la variable externa tomará el valor 0.

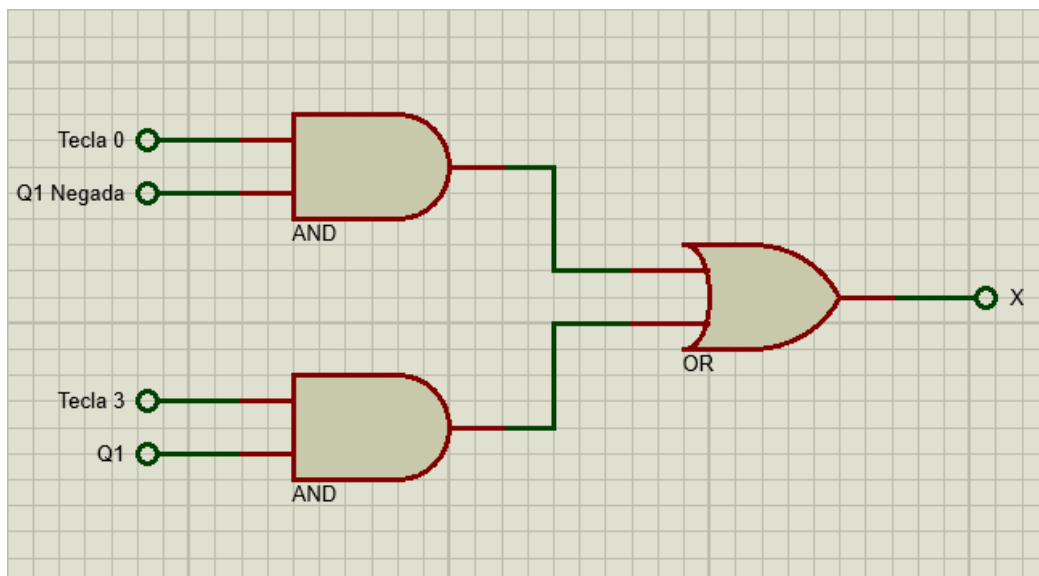


Imagen 3: Circuito Combinacional de Variable Externa

Clock:

Por consigna, el clock debe ser generado por el Teclado Matricial, pero al realizarlo tuvimos que tener en cuenta el retardo eléctrico generado por utilizar codificadores, comparadores y luego circuitos combinacionales para definir el valor de la variable externa. Para solucionar esto decidimos utilizar un clock invertido, o sea que se mantenga constante en 1 si no se presiona ninguna tecla, y al presionar baje a 0, para que al soltar la tecla vuelva a subir y se efectúen los cambios en los Flip-Flops, con la variable externa ya definida en el valor correspondiente.

Esto es posible debido a que utilizamos Flip Flops JK de Acción Ascendente (solo transiciona cuando el valor de Clock pasa de "LOW" a "HIGH").

Para lograr dicho cometido, utilizamos una compuerta NOR previa al contacto con los Flip-Flops.

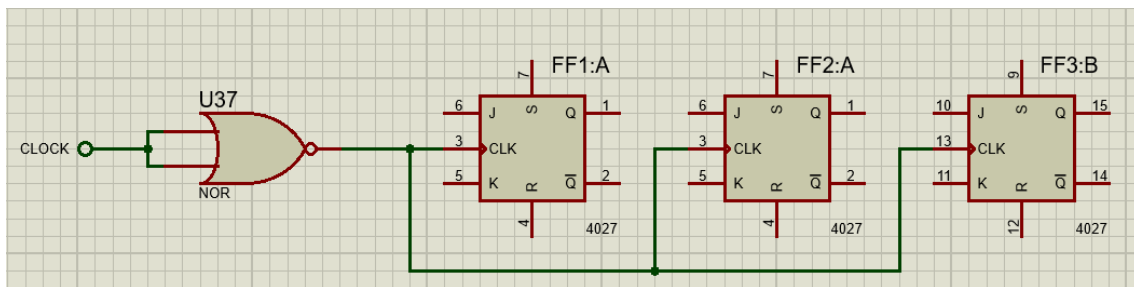
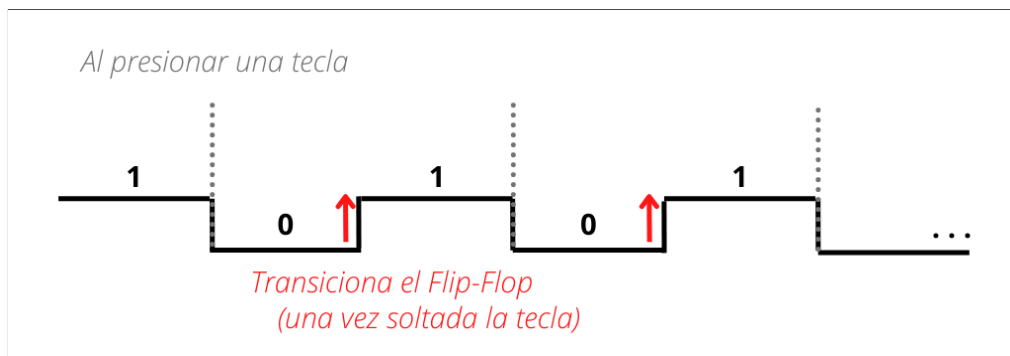


Imagen 4: Clock



Cuadro 1: Concepto sobre el Funcionamiento del Clock

Se utilizaron 2 resistencias Pull-Down, ambas de 10 [K Ω].

2. Diagrama de Estados

Partiendo de la consigna de que para realizar el trabajo debemos tomar los estados **E0** como “Estado de Reposo”, **E2** como “Estado Armada”, y **E6** como “Estado Disparada”. En nuestro caso al ser el Grupo 3, nuestro código de Activación es “0-3” y el de Desactivación es “3-0”, bajo el fundamento de que la Variable Externa valdrá X cuando se presionen las teclas “3” y “0” en su respectivo orden, y valdrá \bar{X} cuando no respete el orden o cuando se presione alguna otra tecla.

Contando en que necesitaremos un estado intermedio entre cada uno de los estados principales, entonces serán necesarios 6 estados, dejando así 2 estados sin uso

debido a la desigualdad $\text{Cantidad de Estados} \leq 2^{\text{Cantidad de Flip-Flops JK}}$.

(En nuestra configuración elegida, no utilizaremos los estados E1 y E7).

Como se explicó previamente, nuestra variable externa, toma el valor “1” sólo cuando:

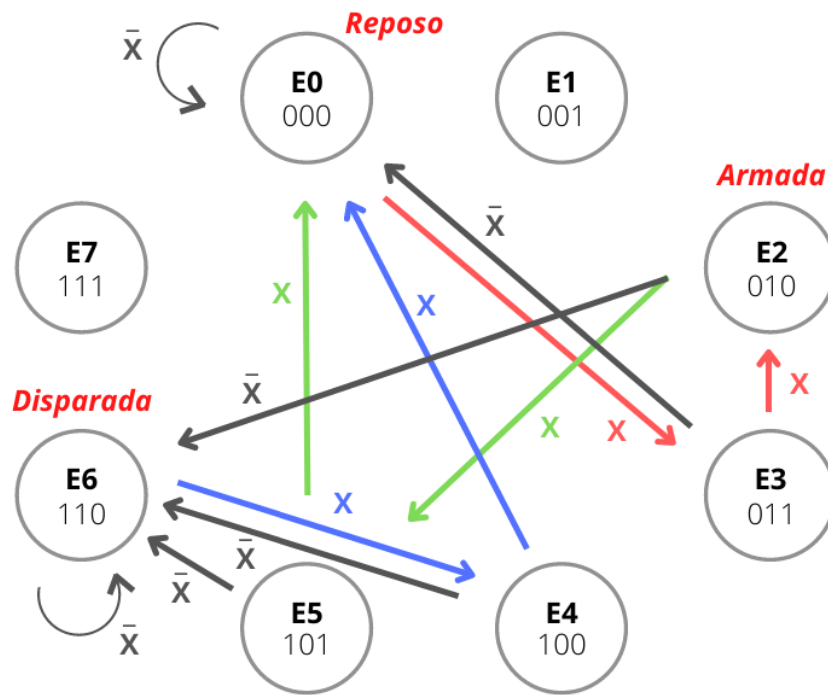
Se presiona la tecla “0” estando en un estado con $Q1 = 0$.

Se presiona la tecla “3” estando en un estado con $Q1 = 1$.

Entonces para generar el diagrama de estados, optamos por utilizar estados cuya $Q1$ varíe de 0 a 1 o viceversa dependiendo del momento en que se encuentre la alarma.

Por ej, si queremos activarla (pasar del estado E0 a E2, utilizaremos el estado E3 como intermedio, debido a que su valor de $Q1$ es igual a 1, entonces pasamos de un estado cuya $Q1$ es 0 (E0) y apretando el primer dígito del código de activación (0) se pasa al estado E3, cuya $Q1$ es igual a 1, entonces para pasar al estado E2 y así activar la alarma, se requiere utilizar el segundo dígito del código de activación, siendo este el número 3, en caso de oprimirse algún otro botón se volverá al estado E0.

La misma lógica se aplica para los siguientes estados, quedando así el diagrama:



Cuadro 2: Diagrama de Estados

3. Tabla de Verdad y de Transiciones

Luego de analizar el diagrama de estados, procedimos a realizar la tabla de verdad y transición del mismo. La última columna corresponde a la conexión de los Flip-Flop para lograr los cambios de estado deseados, comparando con la tabla de transición característica de los Flip-Flops JK.

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Cuadro 3: Tabla de Transiciones de Flip-Flop JK

Estados		t			Variables	t+1			FF J-K					
		Q2	Q1	Q0	X	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
<i>reposo</i>	E0	0	0	0	1	0	1	1	0	X	1	X	1	X
	E1	0	0	1	1	0	0	1	0	X	0	X	X	0
<i>armada</i>	E2	0	1	0	1	1	0	1	1	X	X	1	1	X
	E3	0	1	1	1	0	1	0	0	X	X	0	X	1
	E4	1	0	0	1	0	0	0	X	1	0	X	0	X
	E5	1	0	1	1	0	0	0	X	1	0	X	X	1
<i>disparada</i>	E6	1	1	0	1	1	0	0	X	0	X	1	0	X
	E7	1	1	1	1	1	1	1	X	0	X	0	X	0
<i>reposo</i>	E0	0	0	0	0	0	0	0	0	X	0	X	0	X
	E1	0	0	1	0	0	0	1	0	X	0	X	X	0
<i>armada</i>	E2	0	1	0	0	1	1	0	1	X	X	0	0	X
	E3	0	1	1	0	0	0	0	0	X	X	1	X	1
	E4	1	0	0	0	1	1	0	X	0	1	X	0	X
	E5	1	0	1	0	1	1	0	X	0	1	X	X	1
<i>disparada</i>	E6	1	1	0	0	1	1	0	X	0	X	0	0	X
	E7	1	1	1	0	1	1	1	X	0	X	0	X	0

Cuadro 4: Tabla de Verdad y Transiciones Flip-Flops

Q2	Q1	Q0	LED R	LED A	LED V
0	0	0	0	0	1
0	0	1	X	X	X
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	X	X	X

Cuadro 5: Tabla de Verdad LEDs

4. Mapas de Karnaugh

Posteriormente de realizar la tabla de verdad, procedimos a realizar los mapas de Karnaugh correspondientes a cada entrada de los Flip-Flop JK y cada LED.

J2					K2				
$xQ_2 \setminus Q_1Q_0$	00	01	11	10	$xQ_2 \setminus Q_1Q_0$	00	01	11	10
00	0	0	0	1	00	X	X	X	X
01	X	X	X	X	01	0	0	0	0
11	X	X	X	X	11	1	1	0	0
10	0	0	0	1	10	X	X	X	X

J1					K1				
$xQ_2 \setminus Q_1Q_0$	00	01	11	10	$xQ_2 \setminus Q_1Q_0$	00	01	11	10
00	0	0	X	X	00	X	X	1	0
01	1	1	X	X	01	X	X	0	0
11	0	0	X	X	11	X	X	0	1
10	1	0	X	X	10	X	X	0	1

J0					K0				
$xQ_2 \setminus Q_1Q_0$	00	01	11	10	$xQ_2 \setminus Q_1Q_0$	00	01	11	10
00	0	X	X	0	00	X	0	1	X
01	0	X	X	0	01	X	1	0	X
11	0	X	X	0	11	X	1	0	X
10	1	X	X	1	10	X	1	1	X

Cuadro 6: Mapas de Karnaugh J-K

LED Rojo					LED Amarillo					LED Verde				
$Q_2 \setminus Q_1Q_0$	00	01	11	10	$Q_2 \setminus Q_1Q_0$	00	01	11	10	$Q_2 \setminus Q_1Q_0$	00	01	11	10
0	0	X	0	0	0	0	X	0	1	0	1	X	1	0
1	1	0	X	1	1	0	1	X	0	1	0	0	X	0

Cuadro 7: Mapas de Karnaugh LEDs

Cálculos

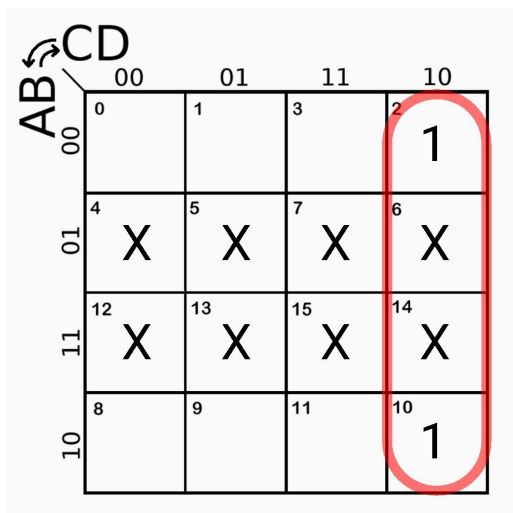
1. Simplificación y armado de Ecuaciones Lógicas

Luego de obtener los Mapas de Karnaugh, seguimos con la agrupación y simplificación de las funciones booleanas de cada diagrama, bajo las siguientes pautas:

- Solo se pueden armar grupos de tamaño 2^n .
- Cada grupo debe ser de mayor tamaño posible, teniendo la posibilidad de superponerse con otro grupo para lograrlo.
- Los valores X (sin cuidado) pueden tomar valor tanto de 0 como 1 dependiendo de su utilidad al cálculo algebraico.

Conexiones Flip-Flop JK: ($A = "X"$, $B = "Q_2"$, $C = "Q_1"$, $D = "Q_0"$)

- **Entrada J2:**



A 4x4 Karnaugh map for variables A, B, C, and D. The columns are labeled CD (00, 01, 11, 10) and the rows are labeled AB (00, 01, 11, 10). The cells contain the following values: (0,0)=0, (0,1)=1, (0,2)=3, (0,3)=2 (circled in red with a '1'); (1,0)=4 (X), (1,1)=5 (X), (1,2)=7 (X), (1,3)=6 (X); (2,0)=12 (X), (2,1)=13 (X), (2,2)=15 (X), (2,3)=14 (X); (3,0)=8, (3,1)=9, (3,2)=11, (3,3)=10 (circled in red with a '1').

AB \ CD	00	01	11	10
00	0	1	3	2 1
01	4 X	5 X	7 X	6 X
11	12 X	13 X	15 X	14 X
10	8	9	11	10 1

$$J_2 = Q_1 \cdot \overline{Q_0}$$

- **Entrada K2:**

		CD			
		00	01	11	10
AB	00	0 X	1 X	3 X	2 X
	01	4	5	7	6
	11	12 1	13 1	15	14
	10	8 X	9 X	11 X	10 X

$$K_2 = x \cdot \overline{Q_1}$$

- **Entrada J1:**

		CD			
		00	01	11	10
AB	00	0	1	3 X	2 X
	01	4 1	5 1	7 X	6 X
	11	12	13	15 X	14 X
	10	8 1	9	11 X	10 X

$$J_1 = \overline{x} \cdot Q_2 + x \cdot \overline{Q_2} \cdot \overline{Q_0} = \overline{\overline{x} \cdot Q_2} = \overline{\overline{x}} + \overline{Q_2} = x + \overline{Q_2} + x \cdot \overline{Q_2} \cdot \overline{Q_0}$$

(Aclaración: se usó el teorema de DeMorgan en el primer término debido a que en el circuito físico un chip NOR trae 4 compuertas, entonces para hacer uso de las mismas y no solo para el CLOCK y Sensor, reduciendo el uso de chips AND o OR a las ecuaciones que poseían \overline{X} en las mismas las desarrollamos para evitar su uso).

- **Entrada K1:**

AB ↻ CD		00	01	11	10
		0	1	3	2
00	X	X	1		
01	X	X			
11	X	X		1	
10	X	X		1	

$$K_1 = x \cdot \overline{Q_0} + \overline{x} \cdot \overline{Q_2} \cdot Q_0 = x \cdot \overline{Q_0} + \overline{x} + Q_2 \cdot Q_0$$

(Misma aclaración que arriba)

- **Entrada J0:**

AB ↻ CD		00	01	11	10
		0	1	3	2
00		X	X		
01	4	X	X	6	
11	12	X	X	14	
10	8	1	X	X	10

$$J_0 = x \cdot \overline{Q_2}$$

- **Entrada K0:**

		CD			
		00	01	11	10
AB ↻	00	0 X	1 	3 1	2 X
	01	4 X	5 1	7 	6 X
	11	12 X	13 1	15 	14 X
	10	8 X	9 1	11 1	10 X

$$K_0 = x \cdot \overline{Q_2} + Q_2 \cdot \overline{Q_1} + \overline{Q_2} \cdot Q_1$$

Conexiones para los LED: (A = "Q₂", B = "Q₁", C = "Q₀")

- **LED Rojo:**

		BC			
		00	01	11	10
A ↻	0	0 	1 X	3 	2
	1	4 1	5 	7 X	6 1

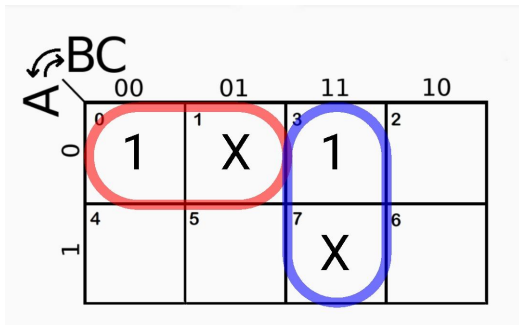
$$LED\ ROJO = Q_2 \cdot \overline{Q_0}$$

- **LED Amarillo:**

		BC			
		00	01	11	10
A ↻	0	0 	1 X	3 	2 1
	1	4 	5 1	7 X	6

$$LED\ AMARILLO = Q_2 \cdot Q_0 + \overline{Q_2} \cdot \overline{Q_0} \cdot Q_1 = Q_2 \cdot Q_0 + \overline{Q_2} \cdot J_2$$

- **LED Verde:**



$$LED\ VERDE = \overline{Q_2} \cdot \overline{Q_1} + Q_1 \cdot Q_0$$

2. Resistencias LEDs

Tomando una tensión promedio de 2 [V] de Voltaje Directo en cada LED.

Tomando una corriente de 20 [mA] de Corriente Directa en cada LED.

$$R_{LEDs} = (V_{Fuente} - V_{LED}) / I = 200 [\Omega]$$

Utilizamos una resistencia de 220 [Ω] ya que es el valor más cercano de resistencia que poseíamos al alcance.

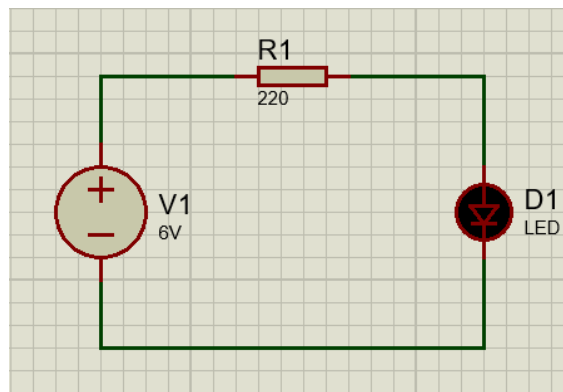


Imagen 6: Diagrama Conceptual a la conexión de cada LED

Diagrama Circuital

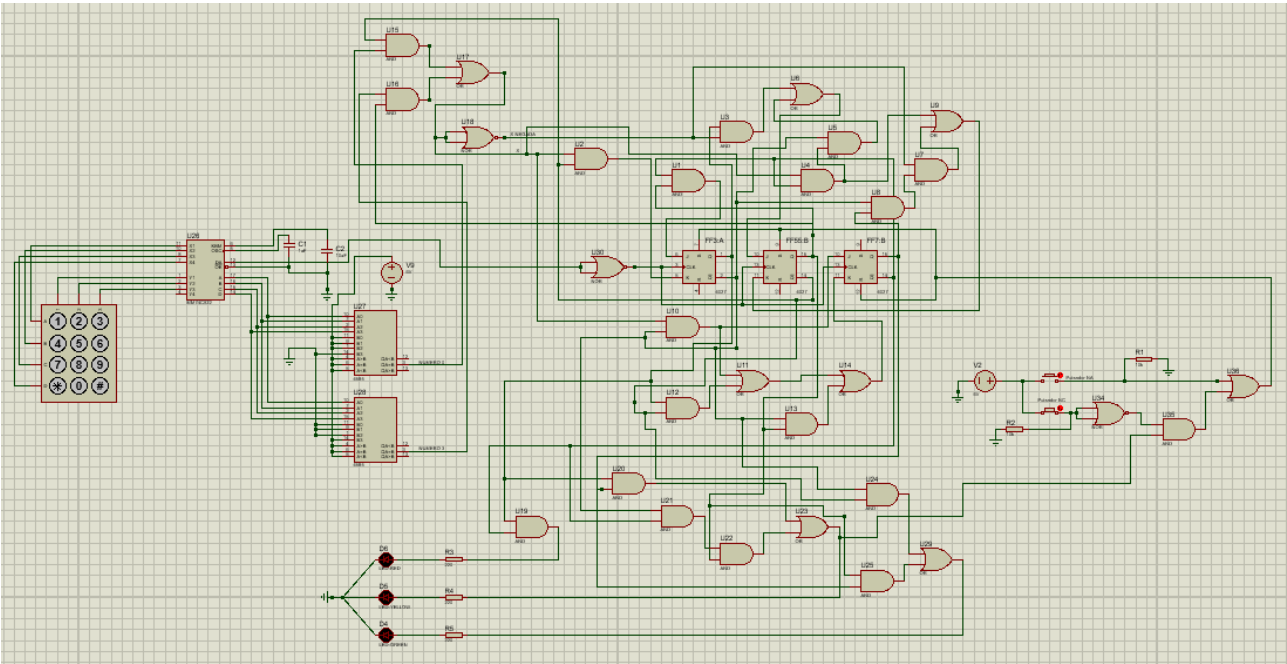


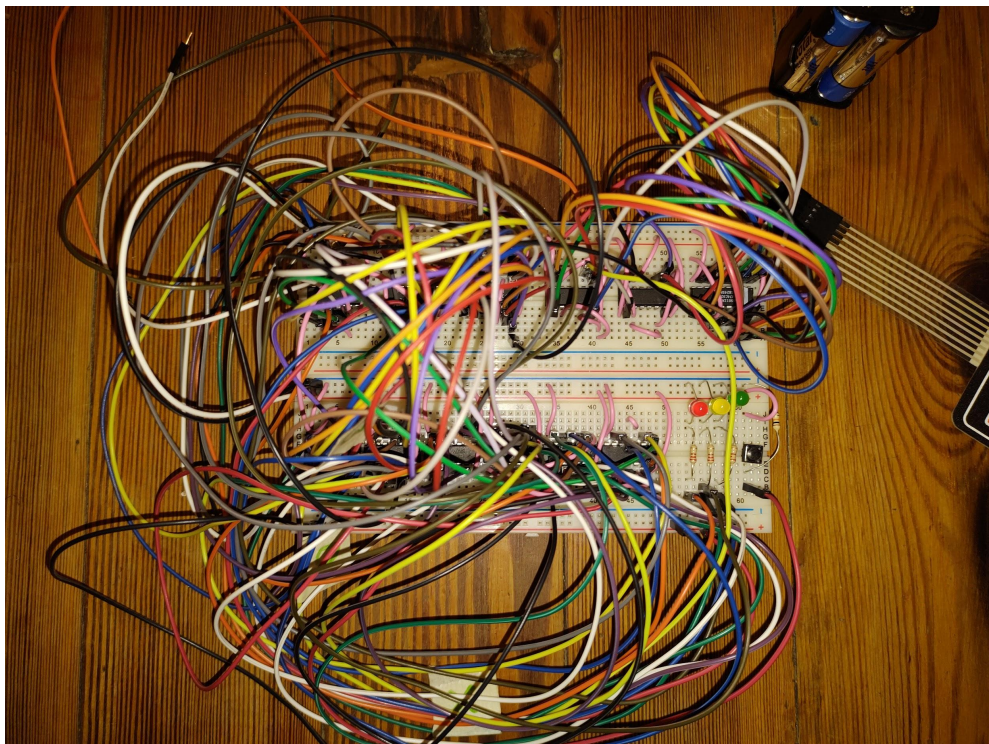
Imagen 7: Diagrama Circuital Alarma en *Proteus*

Materiales

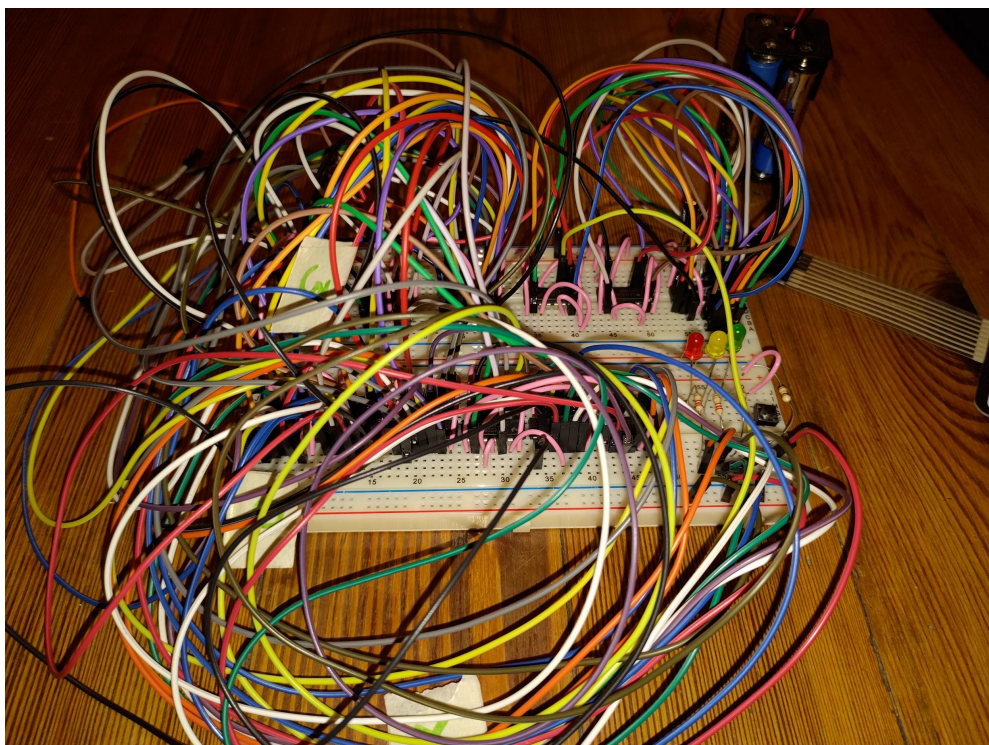
- Fuente de alimentación de 6 V (4 pilas de 1,5 V)
- 1 Capacitor de 10 μF
- 1 Capacitor de 1 μF
- 3 Flip-Flops JK Ascendentes con Set/Reset [2 Chips] (HEF4027B)
- 2 Chips Comparadores 4 bits (CD4585B)
- 4 Chips AND de 4 compuertas (74LS08)
- 2 Chips OR de 4 compuertas (74LS32)
- 1 Chip NOR de 4 compuertas (CD4001BC)
- 1 Teclado Matricial 4x4
- 1 Pulsador NC
- 1 Pulsador NA
- 2 Placas Protoboard
- 1 Chip Decodificador Teclado Matricial (MM74C922)
- 1 LED Verde
- 1 LED Rojo
- 1 LED Amarillo
- 3 Resistencias 220 $[\Omega]$
- 2 Resistencias 10 $[\text{K}\Omega]$

(Data sheets adjuntados al final del informe)

Gráficos Topológicos



Imágenes 8 y 9: Vistas dorsal y lateral del Circuito Físico



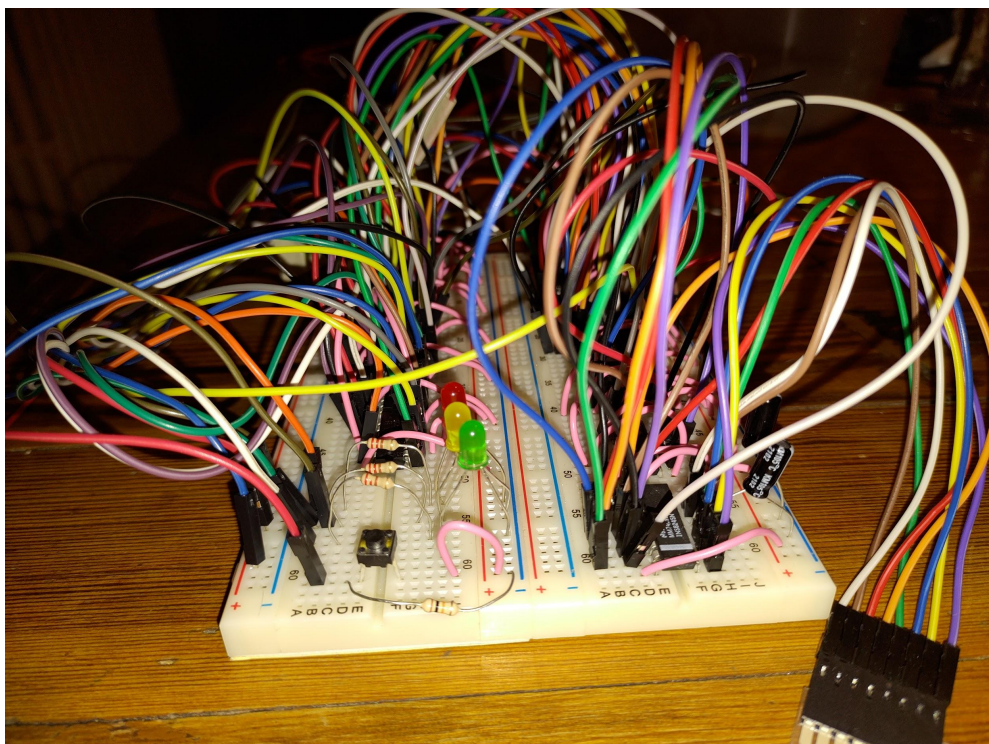


Imagen 10: Vista lateral del Circuito Físico

Conclusiones

En este trabajo práctico se logró construir un autómata de Moore tipo Alarma de 6 estados con Botón de Pánico y Sensor NC con una sola variable externa, aplicando conocimientos sobre el álgebra de Boole, Teorema de DeMorgan, Sistemas Secuenciales, Flip-Flops JK, Sistemas Combinacionales y otros *MSI*, permitiéndonos adquirir y mejorar conocimientos prácticos sobre el armado de circuitos utilizando placas Protoboard.

Destacando los conocimientos previos adquiridos en el TP anterior a la hora de trabajar en Protoboard y compuertas, así como la necesidad de lectura y comprensión de las *datasheet* para la realización del trabajo.

Bibliografía y referencias

- [1] All Data Sheets
<https://www.alldatasheet.es/>

Hojas de Datos