

# Bitácora Proyecto 1 Fundamentos de Arquitectura de Computadores

Rodríguez Montero Juan Daniel

Vega Marín Giancarlo

**13 de agosto:** Asignación del proyecto por parte del profesor.

Compra parcial de componentes para el circuito. Se compra solo transistor y Display de 7 segmentos

**16 de agosto:** Grupo pareja formado para proyecto

**22 de agosto:** Se investiga acerca de la diferencia entre TTL y CMOS, para comprar los módulos correctos de Flip Flop y Decodificador BCD.

**24 de agosto:** Se compra el Flip Flop y BCD en MicroJPM

**27 de agosto:** Junto con el compañero, se logra resolver los mapas de Karnaugh, y se obtiene la vista previa de las compuertas necesarias para ensamblar ambos circuitos para el decodificador 1.

**29 de agosto:** Se aclaran dudas con el profesor sobre la funcionalidad y conexión del Flip Flop a las compuertas. Se descubre que el segundo circuito hecho el pasado 27 de agosto estaba mal diseñado. Entradas faltantes, ya que se omitió las entradas conectadas a las salidas del Flip Flop.

**30 de agosto:** Después de investigar sobre los integrados menos comunes, se llega a la conclusión de que es posible simplificar aún más el circuito del acumulador si se utiliza compuertas como XOR y XNOR, gracias a la simplificación por medio de propiedades de álgebra booleana.

**31 de agosto:** Se realiza la tabla de verdad y mapa de Karnaugh. Se utiliza el sitio web <http://www.32x8.com/index.html> para mayor facilidad a la hora de realizar el mapa de Karnaugh de la tabla de verdad propuesta.

Dedos	Y1 Y0
0000	01
0001	10
0010	11
0011	00
0100	01
0101	10
0110	11
0111	00
1000	01
1001	10
1010	11
1011	00
1100	01
1101	10
1110	11
1111	00

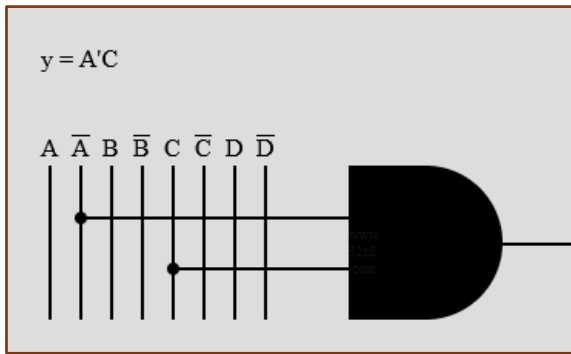
Se propone la primera tabla de verdad, el cual será para el primer circuito combinacional que recibirá los inputs de los primeros 4 switches, los cuales harían la función de los dedos en el proyecto.

En el sitio web anteriormente mencionado, se procede a realizar el mapa K de ambas salidas.

Truth Table					
	A	B	C	D	Y
0	0	0	0	0	x
1	0	0	0	1	0
2	0	0	1	0	x
3	0	0	1	1	1
4	0	1	0	0	x
5	0	1	0	1	x
6	0	1	1	0	x
7	0	1	1	1	1
8	1	0	0	0	x
9	1	0	0	1	x
10	1	0	1	0	x
11	1	0	1	1	x
12	1	1	0	0	x
13	1	1	0	1	x
14	1	1	1	0	x
15	1	1	1	1	0

Siendo la misma tabla de verdad que en la foto previa, se realiza el cálculo y se obtiene el mapa y su suma de productos.

<u>SUM of PRODUCTS</u>				
Map				
	$\overline{C}\overline{D}$	$\overline{C}D$	$C\overline{D}$	$CD$
$\overline{A}\overline{B}$	x	0	1	x
$\overline{A}B$	x	x	1	x
$AB$	x	x	0	x
$A\overline{B}$	x	x	x	x



Se obtiene la compuerta para la salida 1.

Se hace lo mismo para la salida 0 del primer circuito.

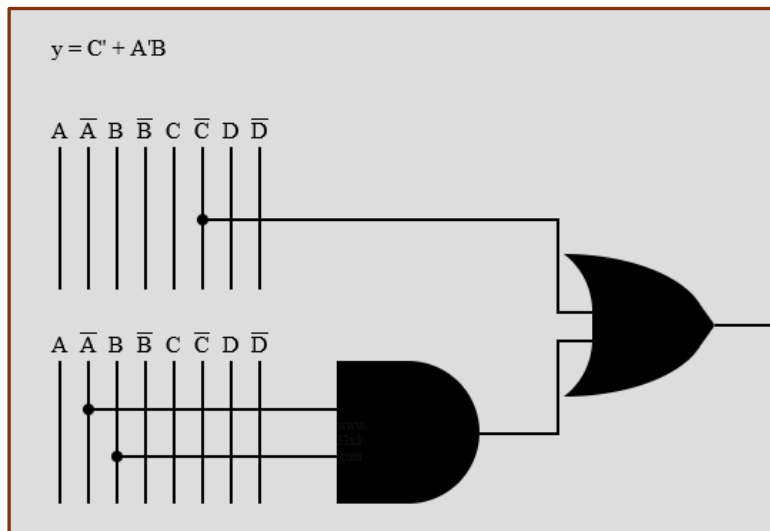
**Truth Table**

	A	B	C	D	Y
0	0	0	0	0	x
1	0	0	0	1	1
2	0	0	1	0	x
3	0	0	1	1	0
4	0	1	0	0	x
5	0	1	0	1	x
6	0	1	1	0	x
7	0	1	1	1	1
8	1	0	0	0	x
9	1	0	0	1	x
10	1	0	1	0	x
11	1	0	1	1	x
12	1	1	0	0	x
13	1	1	0	1	x
14	1	1	1	0	x
15	1	1	1	1	0

## SUM of PRODUCTS

### Map

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$CD$
$\bar{A}\bar{B}$	x	1	0	x
$\bar{A}B$	x	x	1	x
$AB$	x	x	0	x
$A\bar{B}$	x	x	x	x



Confirmando que las salidas de estas compuertas coinciden con la tabla de verdad de la salida 0, se obtienen las compuertas necesarias para el primer circuito combinacional.

Ahora se propone la segunda tabla de verdad, para el circuito combinacional 2, el cual servirá como el acumulador.

$y_1 = C$     $y_0 = D$    Circuito 2  
Acumulador

A B C D	$y_1 y_0$
0 0 0 0	0 0
0 0 0 1	0 1
0 0 1 0	1 0
0 0 1 1	1 1
0 1 0 0	0 1
0 1 0 1	1 0
0 1 1 0	1 1
0 1 1 1	0 0
1 0 0 0	1 0
1 0 0 1	1 1
1 0 1 0	0 0
1 0 1 1	0 1
1 1 0 0	1 1
1 1 0 1	0 0
1 1 1 0	0 1
1 1 1 1	1 0

↓  
2 switches

En la tabla de verdad, además de los dos bits salientes del primer circuito, se tiene como entrada otros 2 bits, los cuales son los últimos 2 switches que harán la función de acumulado (en el proyecto, estos 2 bits provienen de las salidas del registro).

Igual que en el circuito anterior, se realizan los mapas K y suma de productos de ambas salidas.

## Salida 1

### SUM of PRODUCTS

#### Map

	$\overline{C.D}$	$\overline{C}.D$	$C.D$	$C.\overline{D}$
$\overline{A.B}$	0	0	1	1
$\overline{A}.B$	0	1	0	1
$A.B$	1	0	1	0
$A.\overline{B}$	1	1	0	0

#### Groups

(2,3)	$\overline{A.B.C}$
(2,6)	$\overline{A.C.D}$
(8,9)	$A.B.\overline{C}$
(8,12)	$A.C.\overline{D}$
(5)	$\overline{A.B.C.D}$
(15)	$A.B.C.D$

$$y = A'B'C + A'CD' + AB'C' + AC'D' + A'BC'D + ABCD$$

**Truth Table**

	A	B	C	D	Y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

Como se puede apreciar, el resultado de estos cálculos devuelve una suma de productos muy grande, los cuales involucran compuertas de 3 y 4 puertas, donde todas se conectan a múltiples compuertas OR. Esto resultaría en un alto gasto económico, lo cual se propone simplificar con álgebra booleana.

Se procede primero a simplificar las primeras 4 sumas que involucran 3 productos.

$$(\overline{A} \cdot \overline{B} \cdot C) + (\overline{A} \cdot C \cdot \overline{D}) + (A \cdot \overline{B} \cdot \overline{C}) + (A \cdot \overline{C} \cdot \overline{D}) = (A + C) \cdot (\overline{A} + \overline{C}) \cdot (\overline{B} + \overline{D})$$

Se logra simplificar la expresión a una suma de productos más cómoda. Pero al observar detalladamente, se logra apreciar que las primeras dos sumas de productos se asemejan a la propiedad de disyunción exclusiva, utilizada en la compuerta XOR



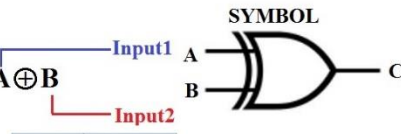
## XOR GATE

### BOOLEAN EXPRESSION

$$A \cdot \bar{B} + \bar{A} \cdot B$$

$$(A + B) \cdot (\bar{A} + \bar{B})$$

C =



INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

ProjectIoT123.com

Esto lleva a la conclusión de que la expresión se puede simplificar aún más al utilizar una compuerta XOR con las entradas A y B.

Volviendo a las dos sumas de productos restantes de 4 productos, se simplifica usando álgebra booleana.

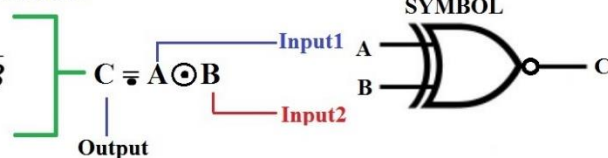
$$(\bar{A} \cdot B \cdot \bar{C} \cdot D) + (A \cdot B \cdot C \cdot D) = B \cdot D \cdot (A + \bar{C}) \cdot (C + \bar{A})$$

Observando detalladamente las sumas de productos, podemos apreciar de que ahora, se encuentra la definición de un XOR negado, o un XNOR.

## XNOR GATE

### BOOLEAN EXPRESSION

$$A + B \cdot \bar{A} + \bar{B}$$



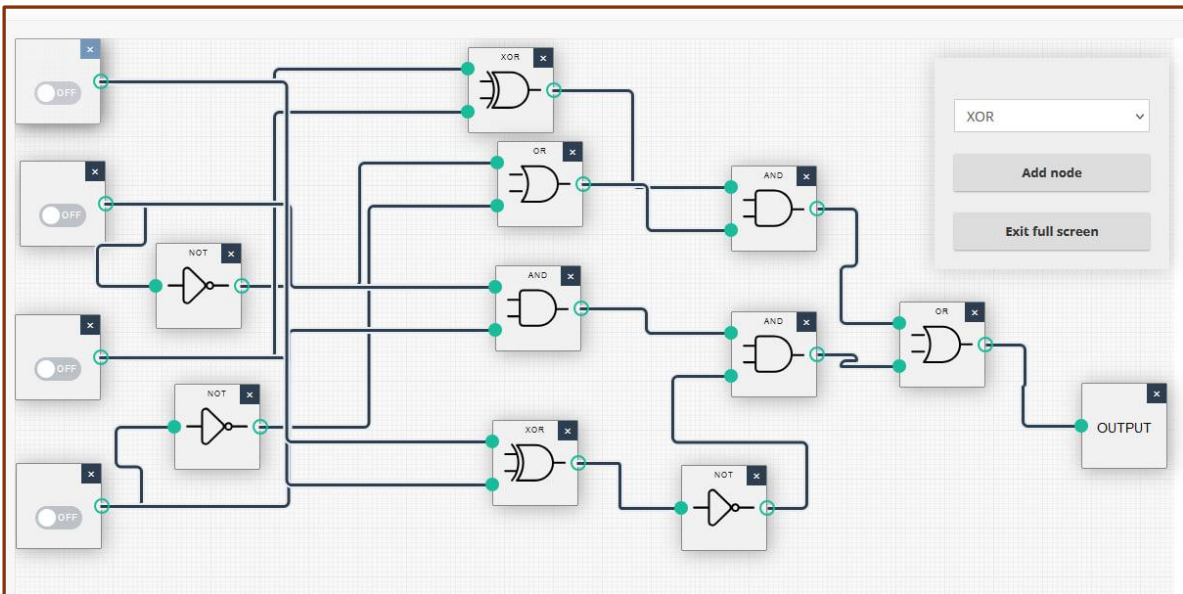
Input		Output
A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

ProjectIoT123.com

Finalmente, se llega a la siguiente expresión:

$$(\overline{B} + \overline{D})(A \oplus C) + (B D)(\overline{A} \oplus C)$$

Se realiza la prueba en un sitio web diseñado para probar compuertas lógicas en línea



Se hacen las pruebas con las entradas de la tabla de verdad del acumulador, y coinciden con las salidas. Por lo tanto, se llega a la conclusión de que la expresión está correcta.

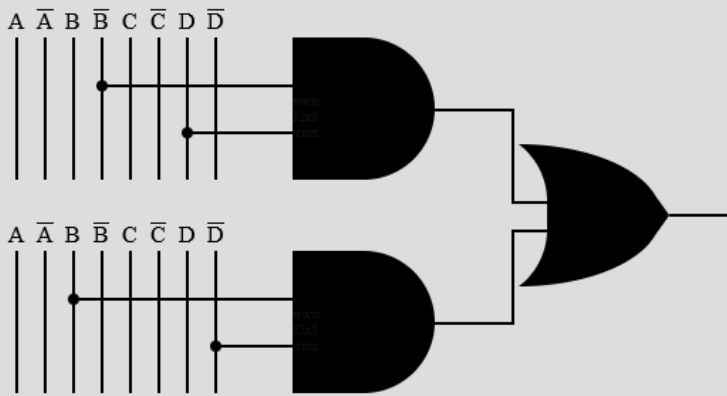
Ahora para finalizar, se realiza el mapa K de la salida 0 del acumulador.

### SUM of PRODUCTS

Map

	$\overline{C}\overline{D}$	$\overline{C}D$	$C\overline{D}$	$CD$
$\overline{A}\overline{B}$	0	1	1	0
$\overline{A}B$	1	0	0	1
$AB$	1	0	0	1
$A\overline{B}$	0	1	1	0

$$y = B'D + BD'$$

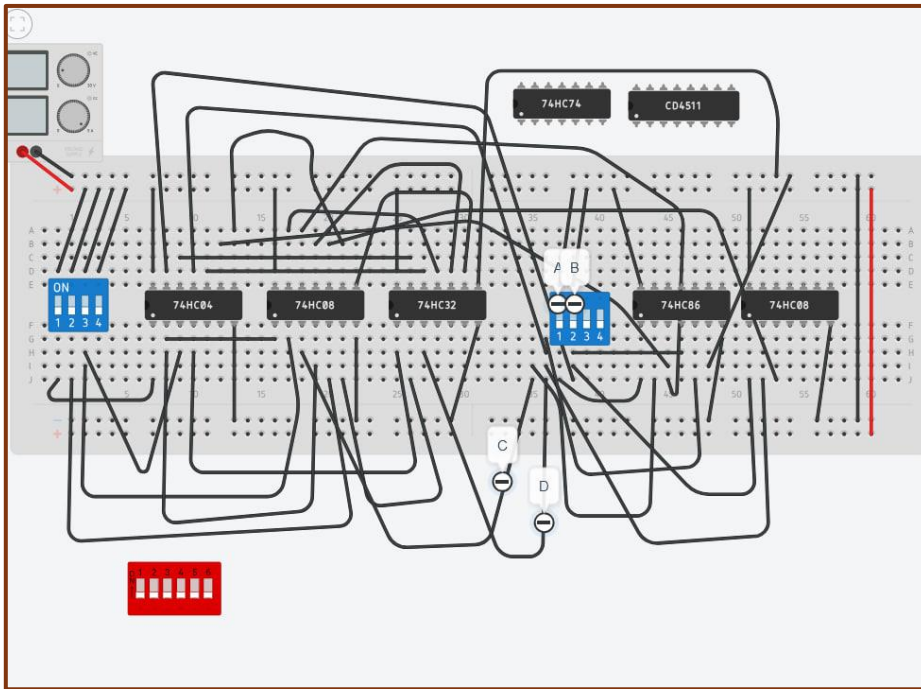


Truth Table					
	A	B	C	D	Y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	0

Si se aprecia bien, se obtiene de nuevo la definición de una compuerta XOR. Y si se observa la tabla de verdad, tiene mucho sentido. Ya que las salidas son 1 cuando B y D son diferentes, y son 0 cuando son iguales.

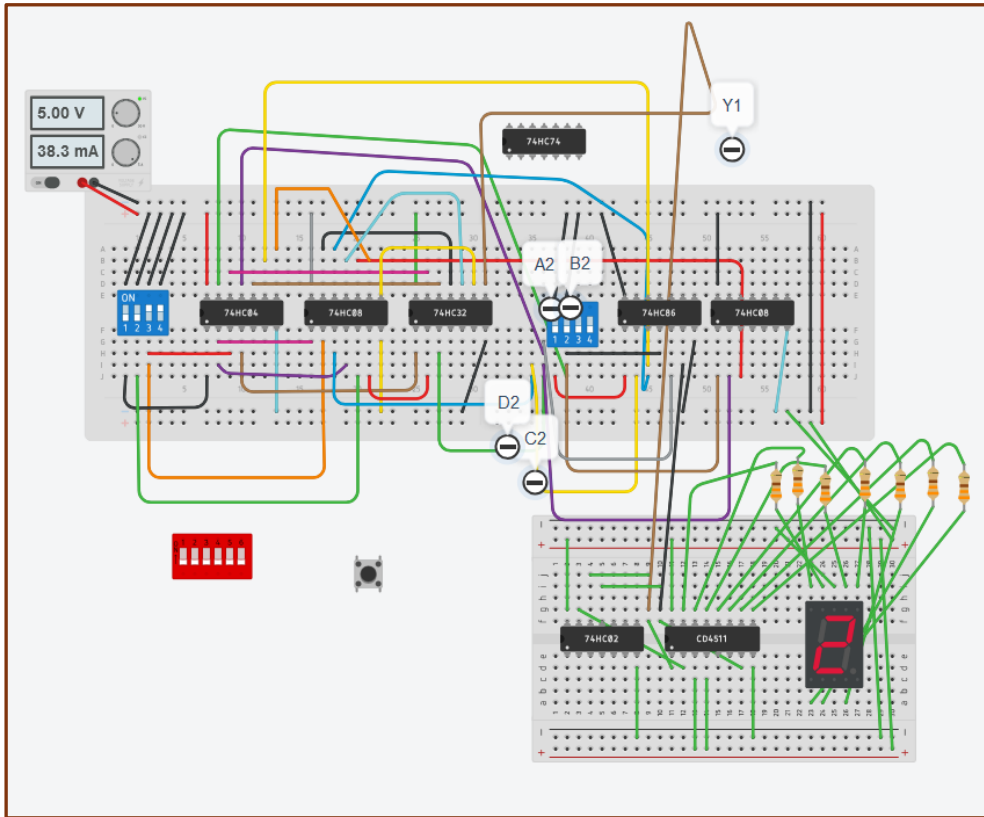
## 2 de septiembre

Se intenta montar el circuito en TinkerCad.



Se conectan los 2 circuitos combinaciones, faltando el registro, BCD y display.

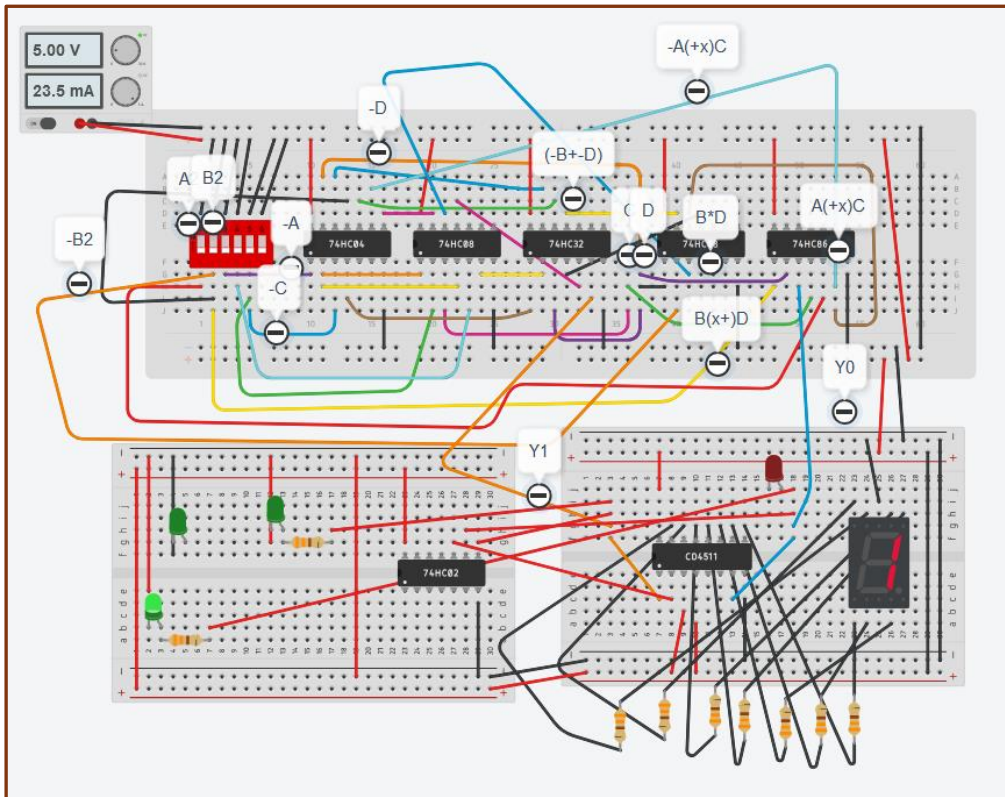
**3 de septiembre**



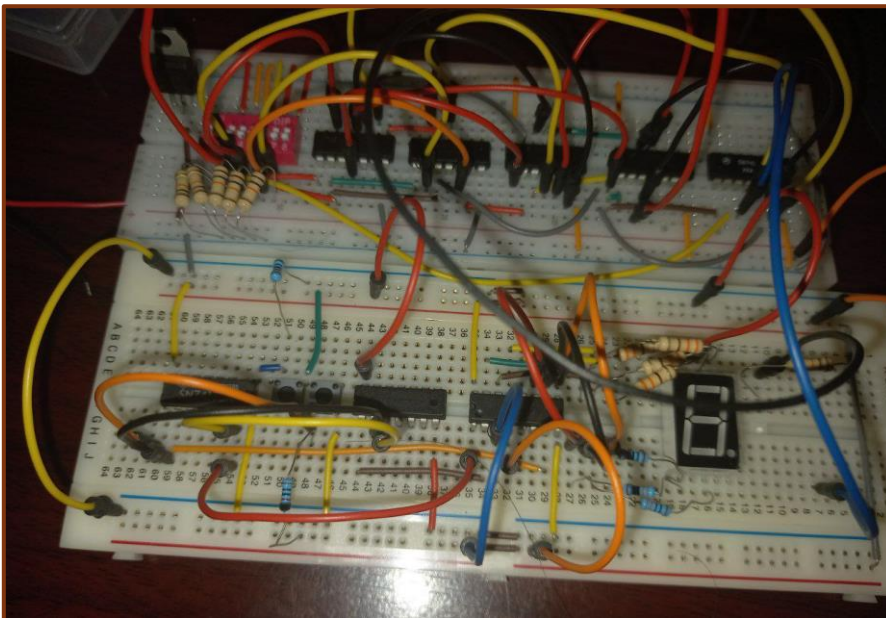
Se conectan las salidas al BCD y se conecta el display de 7 segmentos. Sin embargo, solo funcionan los primeros 4 switches. Los otros 2 switches no suman al resultado existente, lo cual no cumple la función de acumulado.

**4 de septiembre**

Se modifica el circuito en Tinkercad, con un Dip Switch diferente. El acumulado funciona, y el display muestra los resultados correctamente.



Se procede a montar el circuito en la protoboard



Sin embargo, no funciona. El display nunca recibe nada. Se mantiene apagado. A pesar de revisar varias veces las conexiones, no parece que las salidas estén enviando datos lógicos al BCD y por ende al display.

Se prueba con LEDs las salidas, y todas las salidas están incorrectas con respecto a la tabla de verdad establecida.

Se procede a rearmar el circuito en protoboard, pero los resultados siguen fallando. Se llega la conclusión de que existen circuitos quemados, y se tendrán que comprar el siguiente día.

## **5 de septiembre**

La defensa del taller no fue exitosa. Las salidas de la compuerta XOR por alguna razón no mostraban salidas correctas (y no estaba quemado), posiblemente se quemaron 2 BCDs, todos los LEDs se quemaron, el display nunca mostró ningún número. Ni siquiera un 0. Solo se pudo demostrar el trabajo por medio de la bitácora, el circuito montado en la protoboard, y la prueba en TinkerCad.

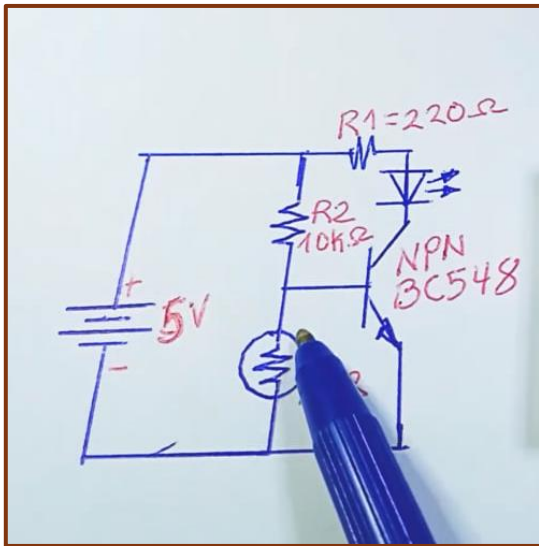
Se llega a la conclusión de que conectado a un pin 5v de Arduino será mejor para el proyecto, ya que un holder de pila 9V es demasiado molesto de implementar. Cualquier otra solución, se buscará llegar a más tardar el fin de semana.

## **8 de setiembre**

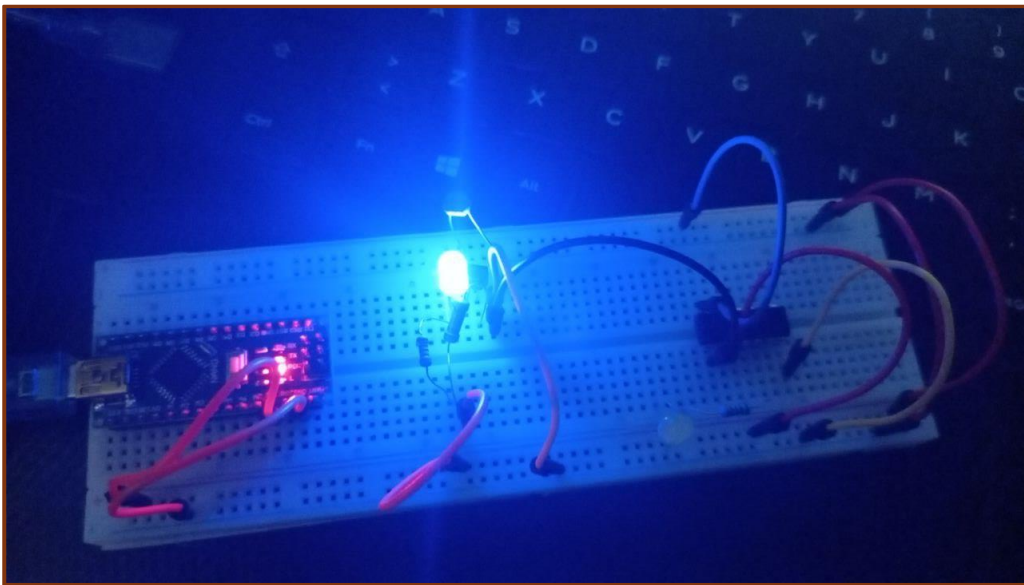
Se procede a reemplazar las pilas de 9V por un Arduino Nano. Se considera un Arduino adicional en caso de que el Nano no logre alimentar todo el circuito, pero aún no se sabe con certeza si eso vaya a pasar.

Se procede a diseñar el circuito que servirá como detector de oscuridad. Para ello, se utiliza el diseño de la siguiente foto:

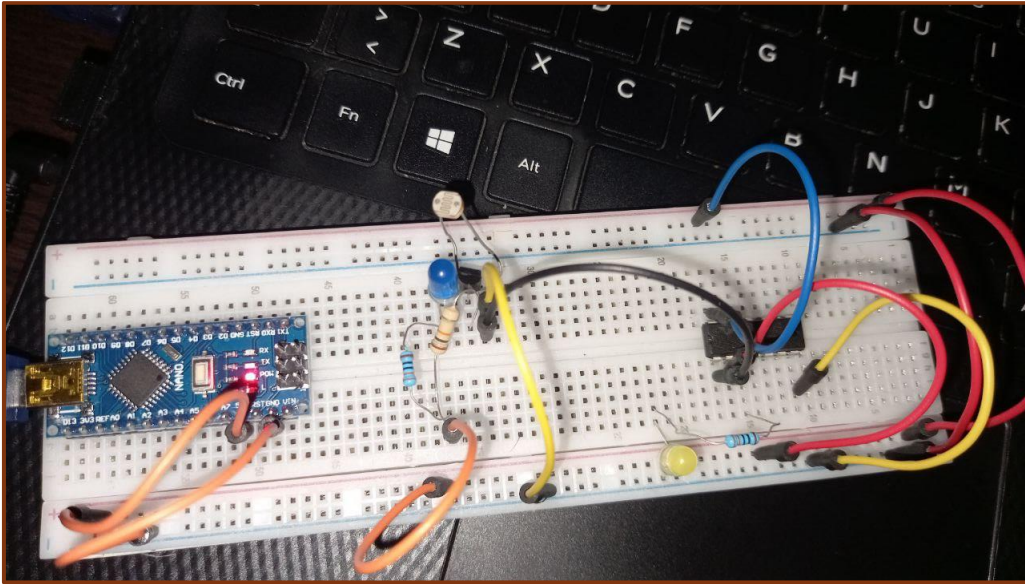




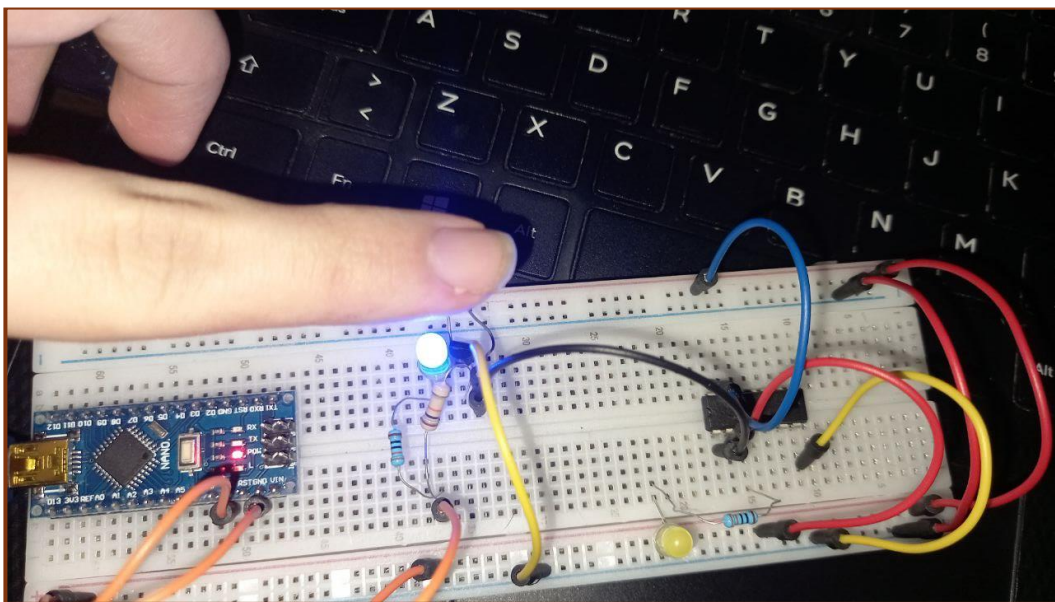
En el caso de nuestro proyecto, se utilizará un transistor NPN 2n2222, y una resistencia de 330 ohm para proteger el diodo LED (para comprobar que funcionará. Después se utilizará para el visualizador)



Se implementa el circuito en la protoboard, alimentado por un Arduino Nano. El LED se prende estando en la oscuridad. Luego se utiliza la linterna del celular para probar que el LED se apague.



Y en efecto, el LED se apaga como se desea. Ahora se tapa el LDR directamente para comprobar su funcionamiento



El paso siguiente será recrear este mismo circuito para todo el arreglo de LEDs y fototransistores requeridos para el funcionamiento del circuito del proyecto.

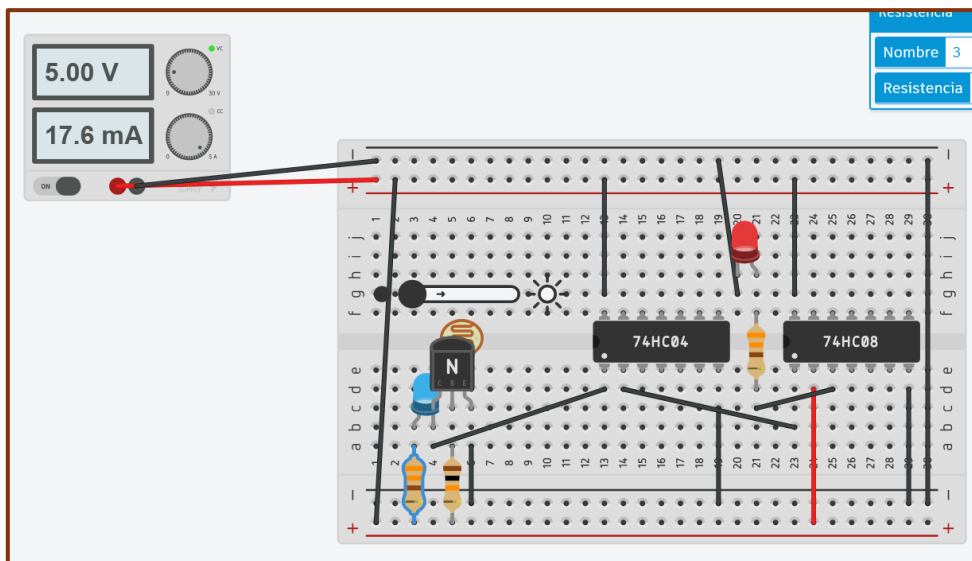
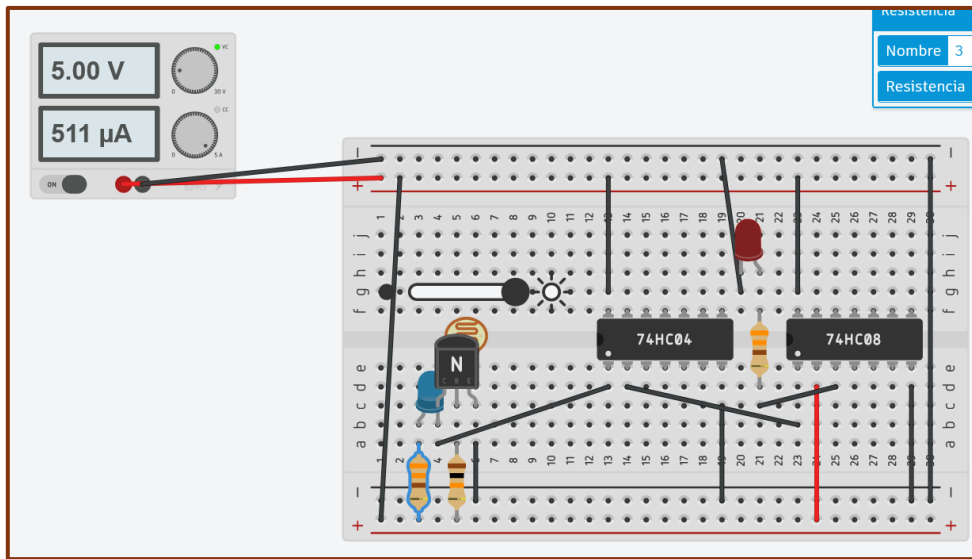
## **9 de setiembre**

Horas después, se decide probar como transferir el 1 y 0 lógico producido por la cantidad de luz detectada, con el fin de que estas sirvan como las entradas del decodificador. Sin embargo, nos encontramos con un problema: el LED de prueba no enciende. Se decide revisar la conexión, y se conecta al cátodo del LED en vez del ánodo, y ahora si prendió. Esto lo encontramos muy curioso, ya que usualmente no sucede esto.

Se decide utilizar una compuerta AND de prueba, donde unas de las entradas se conectan a la fuente Vcc, la otra entrada se conecta al colector del transistor y la salida se conecta a un LED de prueba. En teoría, si el LED enciende por falta de luz, debería haber un 1 lógico que viaje hacia la entrada de la compuerta AND, y por ende la salida será 1 y el LED de prueba encenderá. Sin embargo, ocurrió exactamente lo opuesto. El LED de prueba encendía cuando había luz, y apagaba cuando oscurecía.

Se buscó pruebas que explicaran este comportamiento, y una fuente dudosa decía que el voltaje del colector era pequeño cuando el transistor estaba activo cuando el voltaje de la base era suficiente; en el otro caso, cuando el transistor estaba inactivo, el voltaje del colector era suficiente para enviar un 1 lógico. Se consultará al profesor sobre esto para verificar si es cierto

Simulamos el circuito en TinkerCad, esta vez con una compuerta NOT para verificar si ahora ambos LEDs se prenderían y apagarían al mismo tiempo, lo cual es lo deseado.



Como se aprecia en las capturas, el circuito funciona exactamente que en el de Protoboard, encendiendo ante la falta de luz y apagándose cuando hay luz. Además, ambos LEDs encienden y apagan al mismo tiempo, confirmando la necesidad de un integrado como el 74ls04. Se consultará al profesor si hay otra forma.

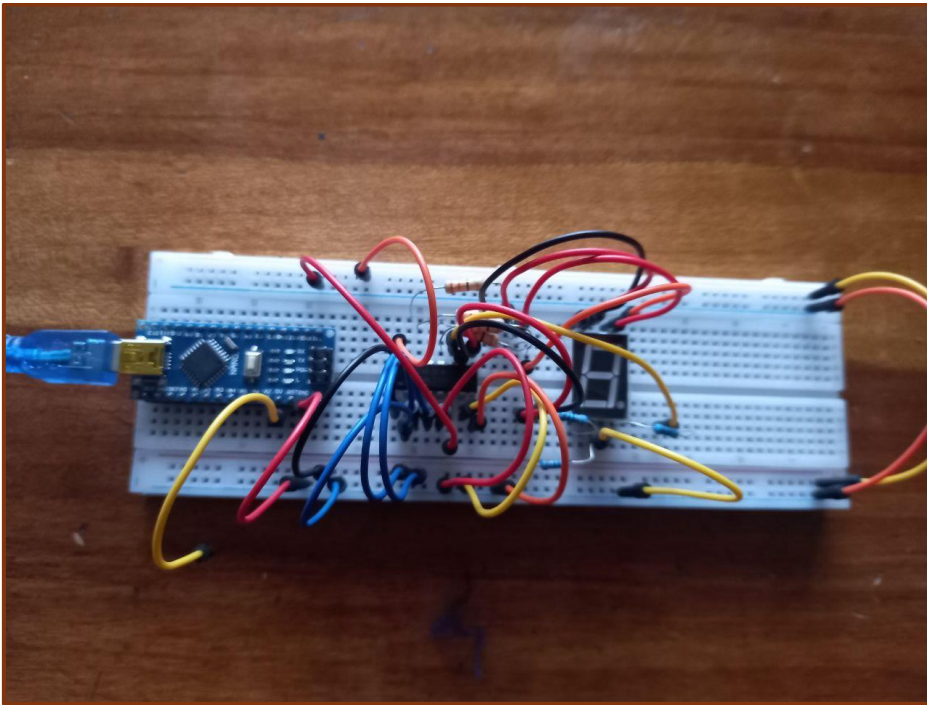
**ACTUALIZACIÓN:** Se confirma con el profesor que la compuerta NOT es la mejor forma de obtener las señales deseadas. Se continúa el diseño y ensamblaje con esto en cuenta.

## 10 de setiembre:

Se compra un nuevo decodificador BCD, con el fin de comprobar de que reciba las salidas del acumulador, y muestre los resultados correctamente.

Después de ensamblar el circuito, se realiza las pruebas en las entradas del integrado BCD, y se comprueba su funcionamiento. Éste estaría listo para ser integrado en el circuito completo.

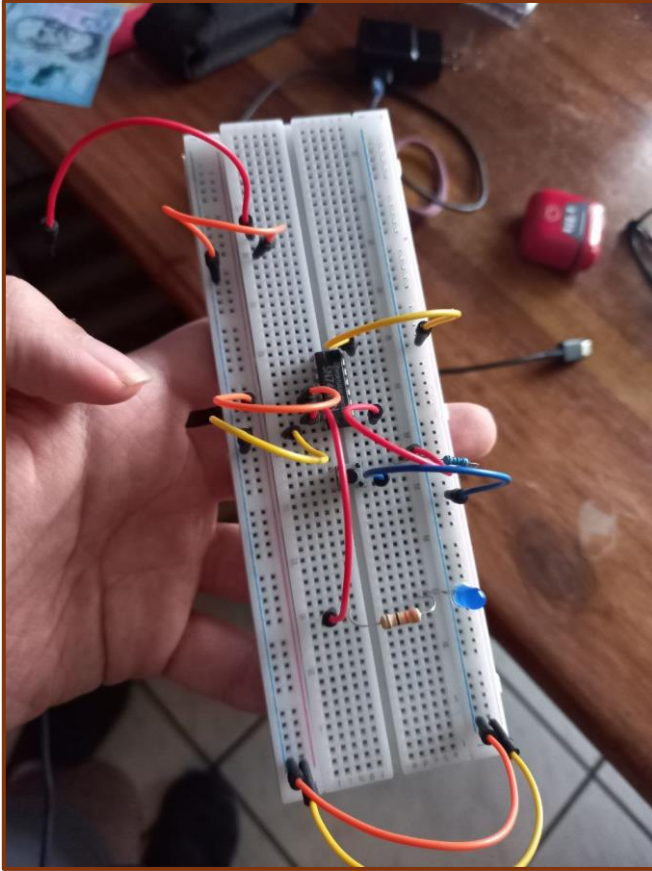
En la siguiente imagen, se ve el diseño parcial del decodificador BCD y display de 7 segmentos.



Además, se procede a hacer el ensamblaje y conexión parcial del registro Flip Flop D, con el fin de comprobar su funcionamiento.

Al finalizar, se realiza una prueba sencilla de cambio de entradas y sus salidas con un botón conectado a una resistencia pull down al pin del CLOCK, con el fin de comprobar su funcionamiento. Sin embargo, los valores de salida no cambian como se desean, al menos que manualmente se cambie el valor del pin clock de LOW a HIGH

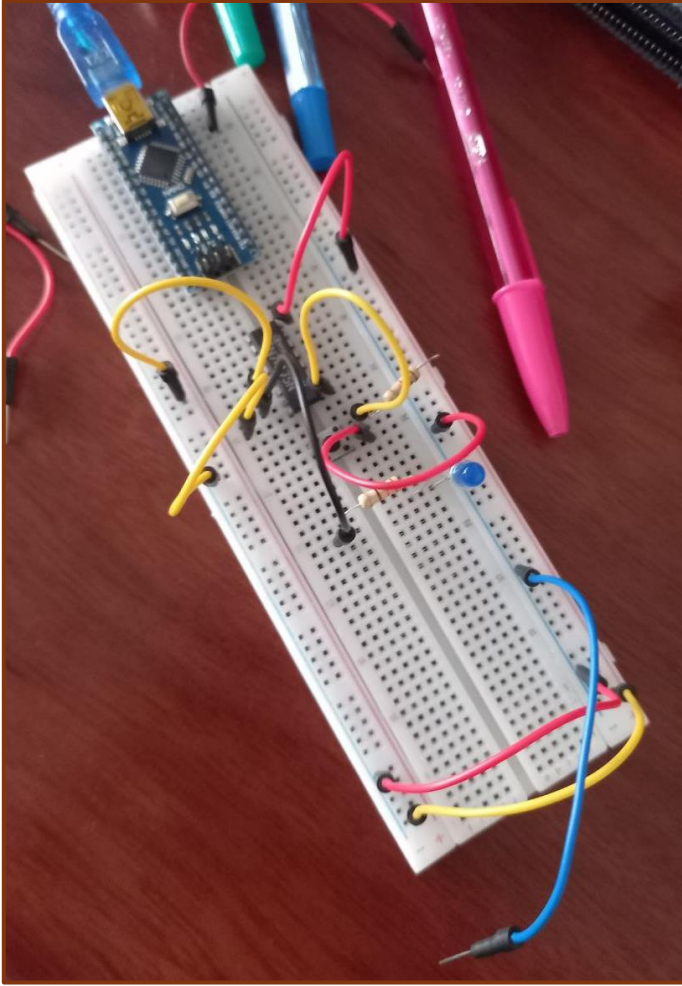




Se llega a la clase con el profesor, consultándole sobre el por qué no funciona. El profesor nos sugiere reducir el valor de la resistencia pull-down, y utilizar el multímetro para comprobar que en efecto está llegando un valor HIGH de voltaje al pin del reloj.

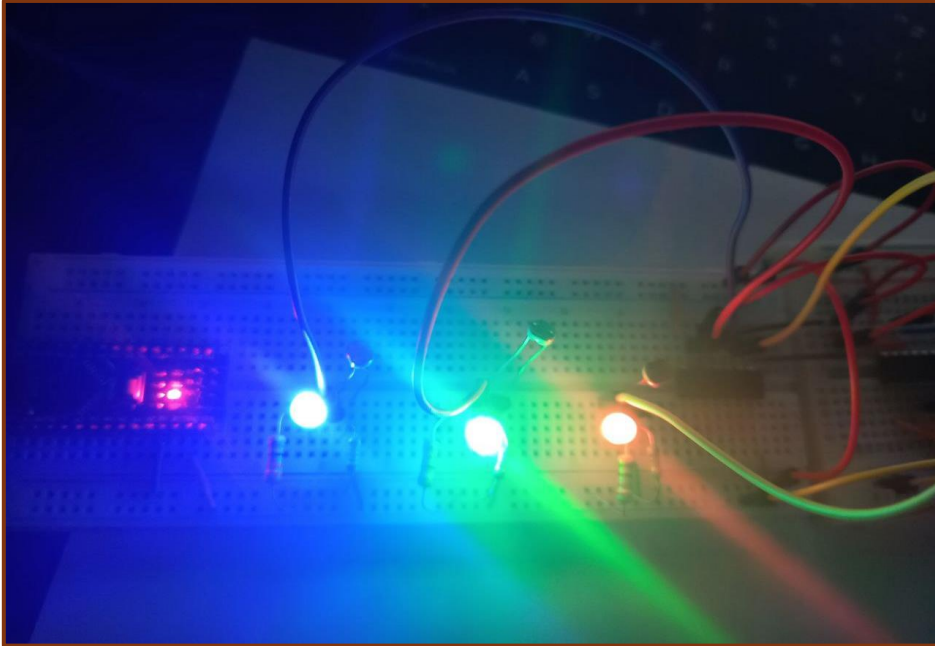
Se hacen esas pruebas, y todo parece en orden, pero aun así no funciona. En un momento de claridad, se decide cambiar la resistencia a una pull-up. Y en efecto, el reloj funcionaba y dejaba pasar el valor de entrada hacia la salida cuando se presionaba el botón.

En la siguiente foto se tiene el diseño final del registro que recibirá las salidas del acumulador.



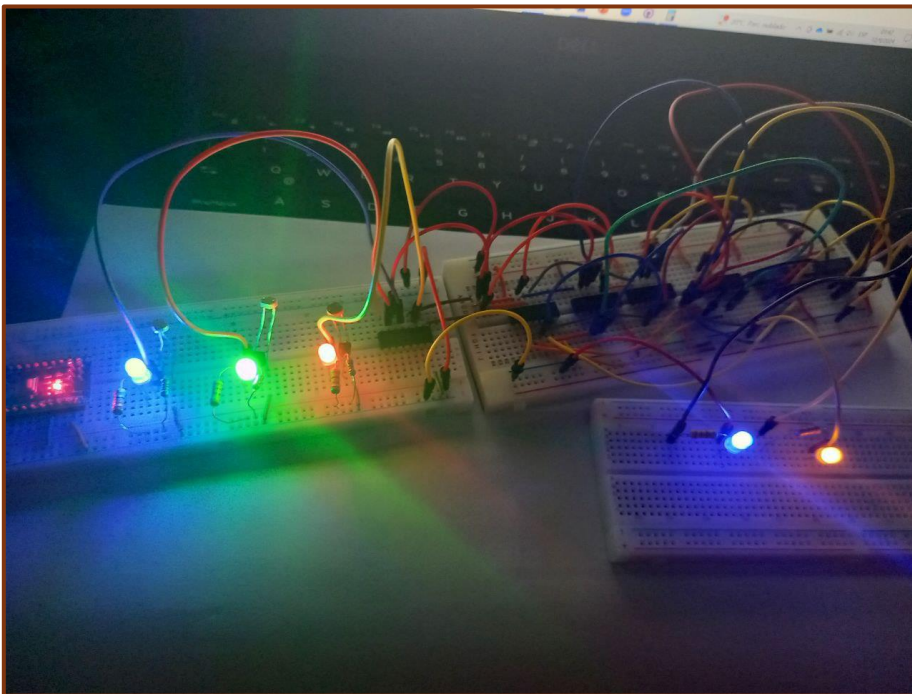
### **11 de setiembre:**

Ya teniendo todo resuelto, se decide armar todo el circuito completo para la defensa del proyecto. Se utiliza las mismas conexiones que se usaron previamente para el detector de oscuridad, y se conectan los 3 LEDs suficientes que servirán como la suma en el acumulador y la muestra en el display



Luego se utiliza la misma conexión de integrados que se utilizó en el taller, solo que esta vez será alimentado por el Arduino Nano, con la esperanza de que esta vez todo el circuito sea alimentado, y la corriente sea suficiente para encender todo.

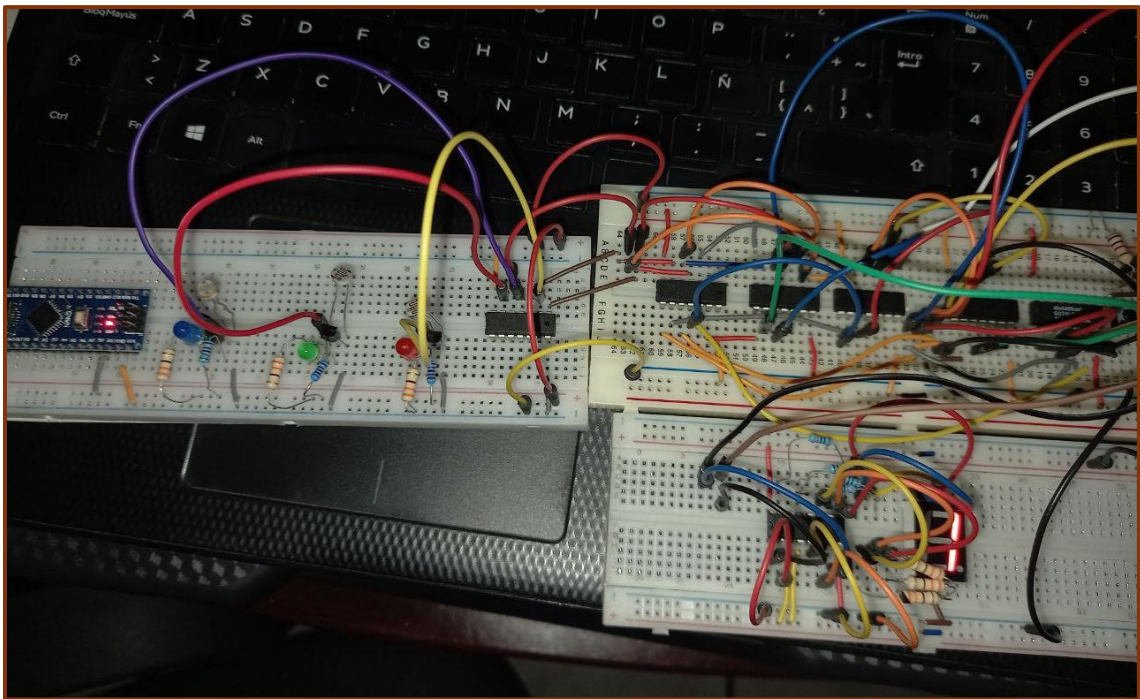
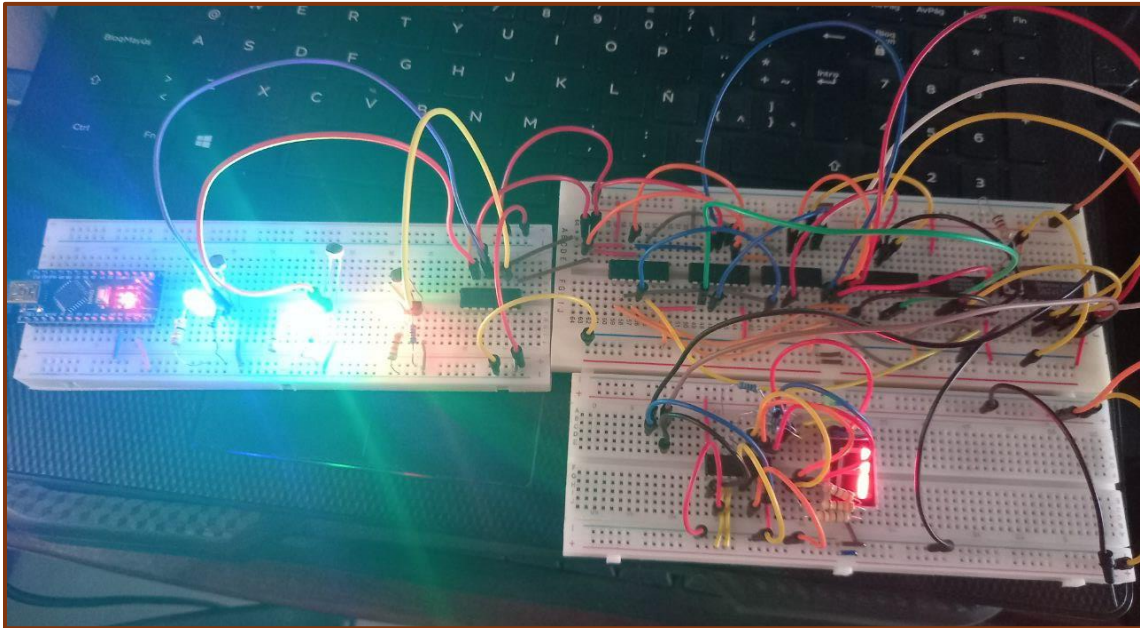
Ambos circuitos combinaciones son ensamblados y conectados exitosamente, y se prueban las salidas iniciales del acumulador con unos LEDs de prueba.





**12 de setiembre:**

Se retoma el ensamblaje del circuito, debido a que aún faltan el registro, el decodificador BCD, display 7 segmentos y el desacople.



Después de un largo proceso, se logra conectar todo, y se comprueba que las conexiones fueron exitosas. Los LEDs prenden y apagan correctamente ante ausencia y presencia de luz

respectivamente, el display de 7 segmentos se actualiza y muestra el resultado correcto cuando se presiona el botón conectado al pin del reloj, y el flanco positivo deja pasar el resultado por medio de sus salidas.

El desacople se realiza en su totalidad en las instalaciones del TEC.

**ACTUALIZACIÓN:** el motor no funcionó. Pero se mostró que el decodificador funcionaba con un LED, por lo que el profesor no nos redujo tantos puntos.

A pesar de unos puntos perdidos, la defensa fue un éxito.