

Actividades miércoles 13 de marzo.

De inicio, se realizó un análisis del código de Gray para saber cuál era la mejor conexión del circuito. Para que la conexión del circuito no fuera tan compleja, se eligió como salida la misma entrada C, como se mira en la tabla 1. De este modo, el circuito solo necesitaría un buffer conectado en la entrada C que brinde como resultado el mismo valor de entrada.

A	B	C	S
0	0	0	0
0	0	1	1
0	1	1	1
0	1	0	0
1	1	0	0
1	1	1	1
1	0	1	1
1	0	0	0

Tabla 1: Código de Gray de 3 bits con su salida a conveniencia.

Para una mejor visualización en que solo importaría la entrada C para generar una salida igual que esta, se realizó un mapa K que mapea el código de Gray y así definir la salida. Como se puede ver en la tabla 2, no importa cuál sea el valor de AB siempre y cuándo C sea 1, por lo que $S = C$.

C \ AB	00	01	11	10
0	0	0	0	0
1	1	1	1	1

Tabla 2: Mapa K del código de Gray

Para implementar esto se investigó sobre cómo utilizar el Arduino para transformar una señal analógica de un potenciómetro en un dato digital, que permita conocer la posición del potenciómetro que oscila entre valores de 0V hasta 5V. Para ello, se utilizó como fuente los 5V del Arduino y una entrada analógica para conectarlo como se observa en la figura 1.

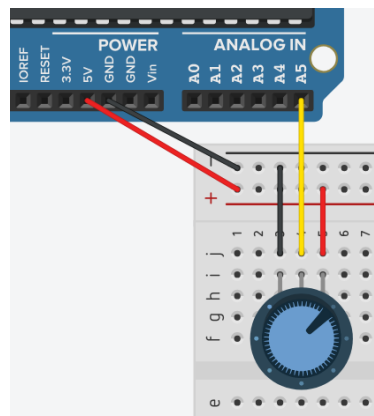


Figura 1: Simulación de la conexión del potenciómetro con el Arduino.

Seguidamente, se definieron rangos del valor digital que retorna la lectura del potenciómetro para establecer el valor resultante de la entrada C. La salida digital máxima que genera el potenciómetro es de 1024, por lo que, si se divide entre 8 posibilidades, cada una tiene un rango de 128. En el código realizado en el IDE de Arduino se estableció una salida digital LOW o HIGH dependiendo de la salida que este genere, es decir, cuándo la salida de la tabla 1 sea 1 se enviará un HIGH, en otro caso se envía un LOW. Esto a través de los rangos definidos anteriormente.

Para la conexión del buffer, se conectó a los 5V del Arduino con su puesta a tierra y a su vez, se conectó la señal digital generada por el Arduino en una entrada y como salida un LED con una resistencia, como se puede observar en la figura 2.

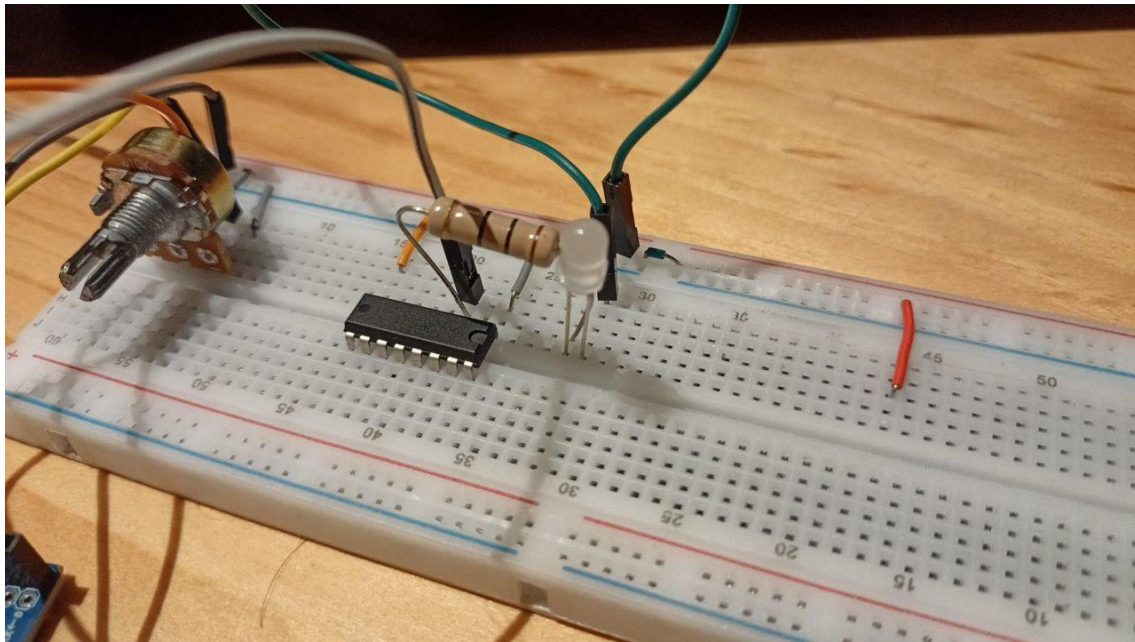


Figura 2: Conexión del buffer.

Ya el Arduino con el código subido y todo conectado, se pudo observar que al mover la perilla del potenciómetro se genera una señal digital que permite encender o apagar el LED por medio del código de Gray.

Actividades lunes 25 de marzo

Durante este día tuve una reunión con algunos de los compañeros del curso por medio de Discord para tener claridad del proyecto. Se trataron temas sobre la elaboración del encodificador y sus mapas K, realizando una comparación de este para verificar si lo que realicé fue correcto. La tabla 3 representa como debe funcionar el encodificador.

A	B	C	S1	S2	S3
0	0	0	0	1	1
0	0	1	1	0	0
0	1	1	1	0	1
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0
1	0	1	0	0	1
1	0	0	0	1	0

Tabla 3: Código de Gray de 3 bits con salida el código de Exceso-3

Para ello se diseñó un mapa K para cada salida, resultando en lo siguiente:

C \ AB	00	01	11	10
0	0	1	1	0
1	1	1	0	0

Tabla 4: Mapa K para el código de Grey con salida S1

C \ AB	00	01	11	10
0	1	1	1	1
1	0	0	0	0

Tabla 5: Mapa K para el código de Grey con salida S2

C \ AB	00	01	11	10
0	1	0	1	0
1	0	1	0	1

Tabla 6: Mapa K para el código de Grey con salida S3

Por medio del Mapa K se obtuvo lo siguiente:

$$S_1 = C\bar{A} + \bar{C}B$$

$$S_2 = \bar{C}$$

$$S_3 = \bar{A}(\overline{B \oplus C}) + A(B \oplus C)$$

La expresión de S_3 no se obtuvo de manera directa, pues hubo que realizar ciertas simplificaciones, como se logra ver en la figura 3.

$$\begin{aligned}
 S_3 &= \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C \\
 &= \bar{A}(\bar{B}\bar{C} + BC) + A(\bar{B}\bar{C} + \bar{B}C) \\
 &= \bar{A}(\overline{B \oplus C}) + A(B \oplus C)
 \end{aligned}$$

* $\overline{B \oplus C}$: XOR negado

Figura 3: Simplificación de S3.

Actividades martes 26 de marzo

Se desarrolló la primera versión de todo el circuito por medio de Tinkercad, incluyendo el circuito combinacional, la visualización de 7 segmentos y el motor, como se puede ver en la figura 4. Además, se desarrolló el código del Arduino. En esta versión todo el circuito funcionó, pero como se puede ver, el circuito tomó mucho material. Este mismo día se buscó un motor DC de un carrito a control remoto antiguo y se comprobó su funcionamiento.

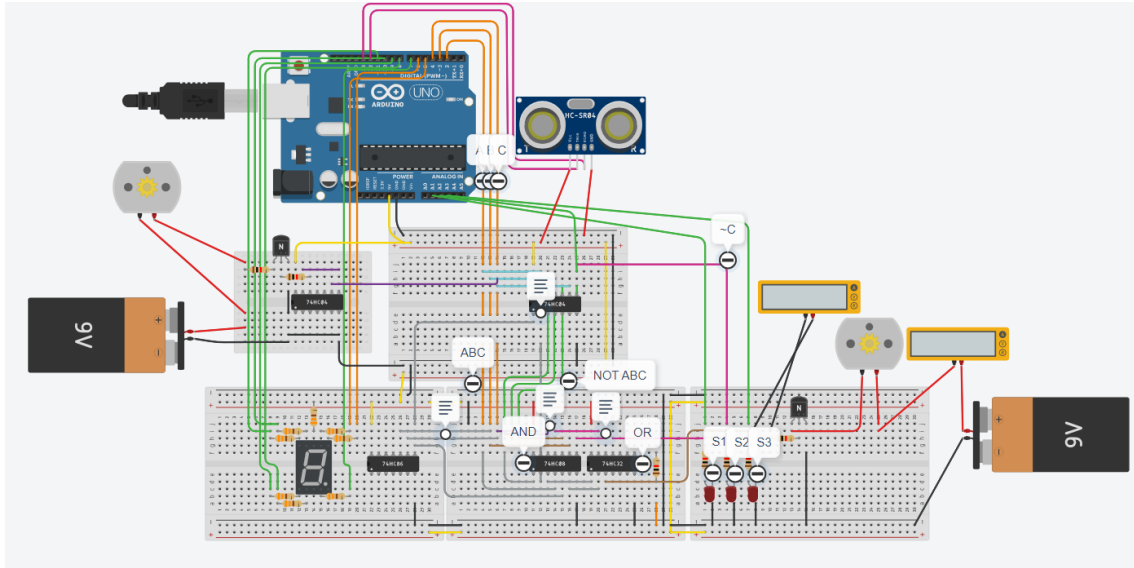


Figura 4: Circuito completo

Actividades miércoles 27 de marzo

Se realizó la compra de los materiales para implementar el circuito a mano. Entre estos materiales están algunas compuertas, cable, un pelador de cable y una protoboard pequeña. Así mismo, se desarrollo una nueva versión del circuito combinacional para que ocupe menos material y sea menos complejo, como se puede ver en la figura 5.

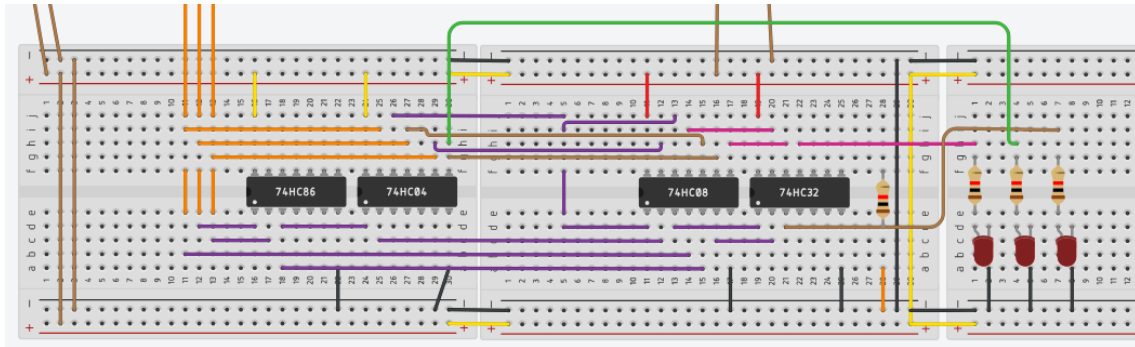


Figura 5: Nueva versión del circuito combinacional.

Este mismo día se estuvo implementando el circuito combinacional en una protoboard.

Actividades jueves 28 de marzo

Se implementó el circuito combinacional en una protoboard. Además, en otras protoboards pequeñas se implementó el 7 segmentos y el motor

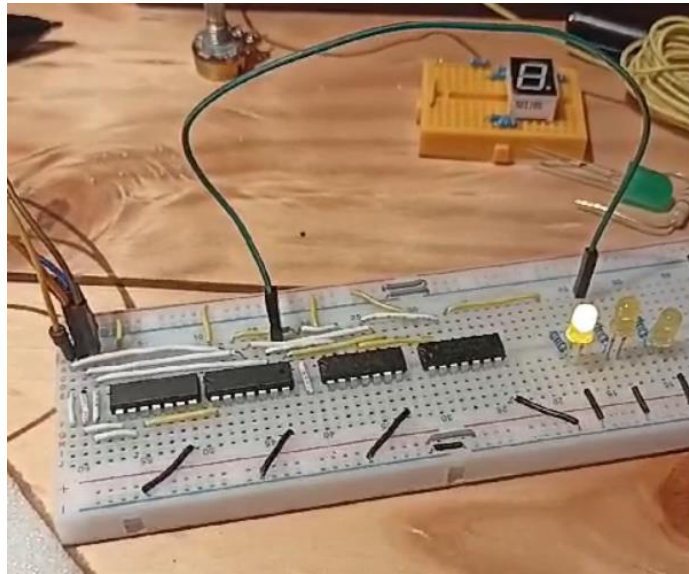


Figura 6: Implementación del circuito combinacional en una protoboard.