



ESCUELA DE INGENIERÍA EN COMPUTADORES

FUNDAMENTOS DE ARQUITECTURA DE COMPUTADORES

CE-1107

**Bitácora: Lógica Combinatoria: Calculadora tomógrafo**

*Sergio Salazar Núñez, 2022155923*

*Daniel Duarte Cordero, 2022012866*

*Profesor: Luis Alberto Chavarría Zamora*

16 de Agosto de 2023

# 1. Introducción

El objetivo de esta bitácora es proporcionar una visión clara y estructurada del proceso seguido, destacando los avances realizados, los problemas encontrados y las soluciones implementadas. Cada día de trabajo está debidamente registrado, incluyendo tanto los aspectos teóricos como las decisiones prácticas tomadas para asegurar el correcto funcionamiento del sistema. Además, se detallan las herramientas empleadas para el diseño automatizado, como software de simulación y simplificación de funciones lógicas, y se validan los procedimientos utilizados a lo largo del desarrollo. Esta bitácora también refleja la evolución del proyecto desde sus primeras etapas de planificación hasta la finalización de la implementación, resaltando los hitos clave alcanzados.

Este registro no solo facilita el seguimiento del progreso, sino que también constituye una herramienta importante para evaluar el proceso de aprendizaje y toma de decisiones a lo largo del proyecto, contribuyendo al desarrollo de competencias en ingeniería y diseño digital. Además, sirve como evidencia de la metodología empleada, permitiendo futuras consultas y mejoras en el diseño propuesto.

## 2. Desarrollo

### 2.1. Tablas Decodificadores 16/8/2024

En esta fase inicial, se diseñó la tabla de verdad para el decodificador encargado de interpretar el número binario ingresado a través de los switches, los cuales simulan la cantidad de dedos levantados. Las entradas de este decodificador están representadas por las señales A, B, C y D, que corresponden a los switches y permiten identificar cuántos dedos están levantados en un momento dado. El decodificador genera las salidas  $Z_1$  y  $Z_0$ , que corresponden al número resultante en formato binario. Este valor binario será utilizado como entrada en la siguiente etapa para la operación de suma.

La tabla de verdad diseñada para el decodificador de dedos es la siguiente:

Cuadro 1: Tabla de verdad decodificador dedos

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	$Z_1$	$Z_0$
1	0	0	0	0	0
1	1	0	0	0	1
1	1	1	0	1	0
1	1	1	1	1	1

Este decodificador sigue un patrón específico en el cual solo ciertas combinaciones de entradas son válidas. Es importante mencionar que, dado que se pretende simular dedos levantados, no todas las combinaciones binarias se consideran válidas, lo cual es reflejado en el diseño de la tabla de verdad.

Posteriormente, una vez obtenida la representación binaria del número de dedos levantados, esta información se utiliza en la operación de suma con el valor acumulado, que proviene de dos switches adicionales. Para esta etapa, se diseñó un segundo decodificador que trabaja con el mismo esquema de entradas binarias A, B, C y D, pero con la característica particular de que su comportamiento es circular. Es decir, al exceder el valor máximo permitido, el resultado rebota y comienza nuevamente desde el valor mínimo. Esto asegura que el sistema opere dentro de un rango predefinido.

Cuadro 2: Tabla de verdad decodificador suma

A	B	C	D	$F_0$	$F_1$
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	1
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	1

La tabla anterior refleja que, cuando el valor acumulado excede el rango establecido, este se reinicia, cumpliendo con la naturaleza circular del sistema, lo cual permite que los resultados nunca excedan el umbral predeterminado.

## 2.2. Mapas de Karnaugh 19/8/2024

Una vez definidas las tablas de verdad de los decodificadores, procedimos a utilizar los mapas de Karnaugh para simplificar las expresiones booleanas que definen las salidas del decodificador. Esto es fundamental para optimizar el diseño del circuito, reduciendo el número de compuertas lógicas necesarias.

A continuación, se presentan los mapas de Karnaugh correspondientes para las salidas  $Z_1$ ,  $Z_0$ ,  $F_0$  y  $F_1$  de los decodificadores.

Cuadro 3: Karnaugh para el  $Z_1$  del deco de Dedos

$\frac{AB}{CD}$	00	01	11	10
00	x	x	0	0
01	x	x	x	x
11	x	x	1	x
10	x	x	1	x

Cuadro 4: Karnaugh para el  $Z_1$  del deco de Dedos

$\frac{AB}{CD}$	00	01	11	10
00	x	x	1	0
01	x	x	x	x
11	x	x	1	x
10	x	x	0	x

Para estos mapas de Karnaugh se marca con x los casos que no se contemplan, ya que el la cantidad de dedos siempre tiene el mismo patrón. Por ejemplo, para que se lea un dedo será necesario hacerlo con el índice extendido, siendo este tomado como 1000. En caso de que se quiera extender el dedo corazón para ser leído como un 1 esto no será posible, ya que el número generado sería 0100, siendo esta una combinación que no se contempla.

Cuadro 5: Karnaugh para el  $f_0$  del deco para la suma

$\frac{AB}{CD}$	00	01	11	10
00	0	1	0	1
01	1	1	0	0
11	0	0	1	1
10	1	0	1	0

Cuadro 6: Karnaugh para el  $f_1$  del deco para la suma

$\frac{AB}{CD}$	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

### 2.3. Simplificación Álgebra Booleana y Simulaciones: 28/8/2024

A partir de los mapas de Karnaugh de la sección anterior se procede a sacar los mintérminos para luego simplificar con álgebra booleana.

$$Z_1 = C \quad (1)$$

$$Z_0 = B\overline{C} + D \quad (2)$$

$$f_0 = \overline{A}B\overline{C} + \overline{A}CD + ACD + ABC + \overline{A}BCD + \overline{A}BCD$$

$$f_0 = \overline{A}\overline{C}(B + D) + AC(B + D)\overline{B}\overline{D}(\overline{A}\overline{C} + \overline{A}C) \text{Asociatividad}$$

$$f_0 = (B + D)(\overline{A}\overline{C} + AC) + \overline{B}\overline{D}(A \oplus C) \text{Disyuncion - exclusiva, Asociatividad} \quad (3)$$

$$f_1 = BD + \overline{B}\overline{D} \quad (4)$$

Estas simplificaciones permiten un diseño más eficiente, reduciendo el número de compuertas lógicas necesarias para implementar los decodificadores.

The diagram shows a logic circuit with four inputs on the left and two outputs on the right. The inputs are connected to a series of logic gates: an OR gate, an XNOR gate, an AND gate, an XOR gate, two NOT gates, and another XNOR gate. The outputs of these gates are connected to two final output nodes, one of which is highlighted in red.

Si bien con esta simulación se reflejaba el comportamiento esperado para la suma circular, se buscó simplificar aún más el circuito con el fin de optimizar el rendimiento y reducir la cantidad de compuertas lógicas requeridas. Teniendo de esta manera la siguiente versión, siendo esta la final

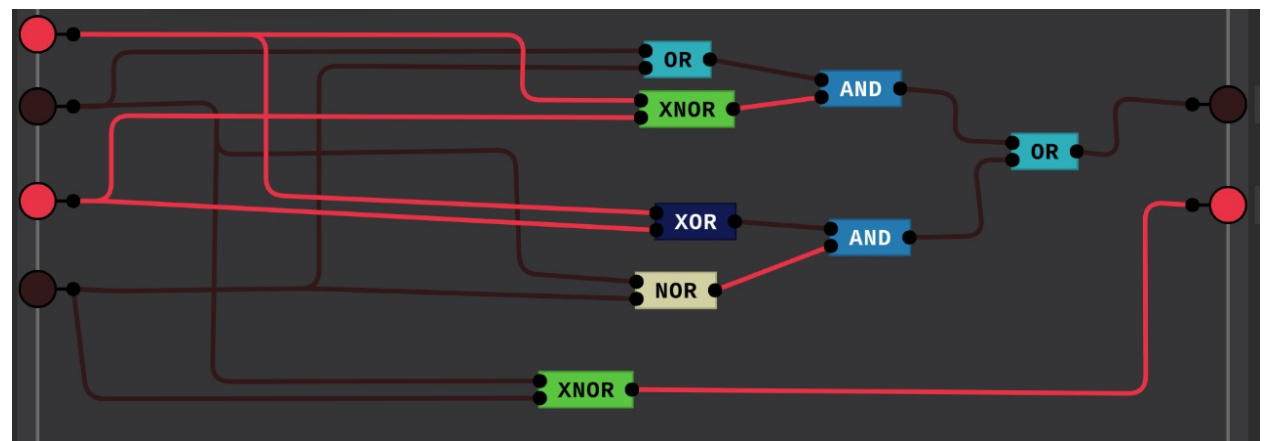


Figura 2.2: Simulación 2 Decodificador Suma

Si bien no se presenta una diferencia significativa entre 2.1 y 2.2. Se logra reducir dos de las compuertas. Teniendo que dos compuertas NOT, ambas conectadas a un AND pueden ser reducidas a un compuerta NOR. Así mismo, se presenta una diferencia en estas imágenes respecto a las ecuaciones 4 y 3. Partiendo de suponer que se tienen dos entradas A y b, al tener  $\overline{A}\overline{B} + AB$  se simplifica a un XNOR, o sea  $A \oplus B$ .

## 2.4. Diagramas de Decodificadores 30/08/2024

Una vez obtenidas las expresiones booleanas simplificadas y las respectivas simulaciones. En las siguientes figuras se muestran los diagramas de los decodificadores implementados para el reconocimiento de los dedos y la suma circular.

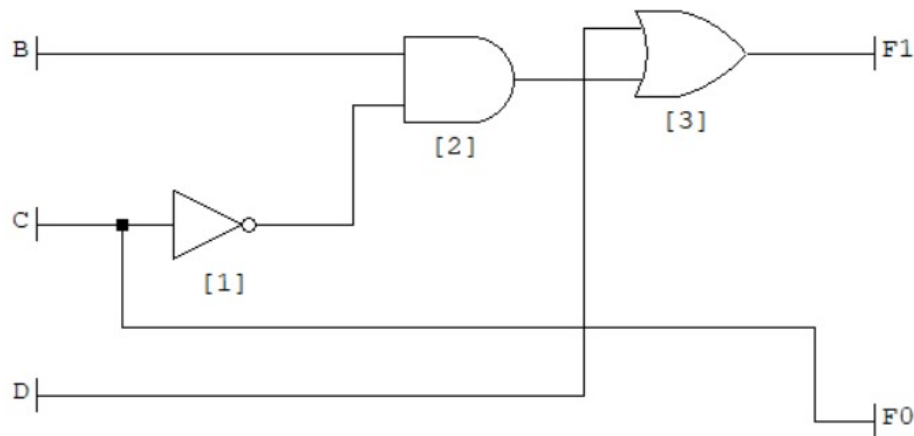


Figura 2.3: Diagrama del decodificador de dedos



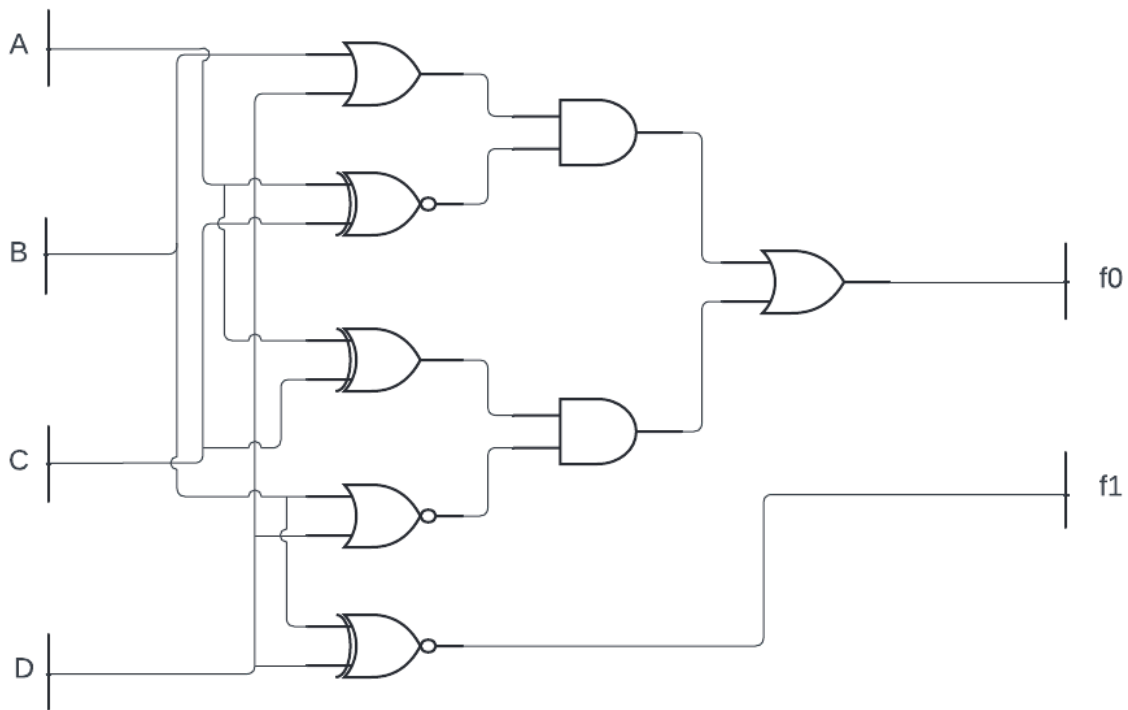


Figura 2.4: Diagrama del decodificador para la suma

## 2.5. Montaje en Protoboard: 2/09/2024

Para el montaje del circuito de lógica digital en protoboard bastó con tomar como referencia los diagramas propuestos para los decodificadores, por lo tanto no se presentó ningún inconveniente en esta etapa.

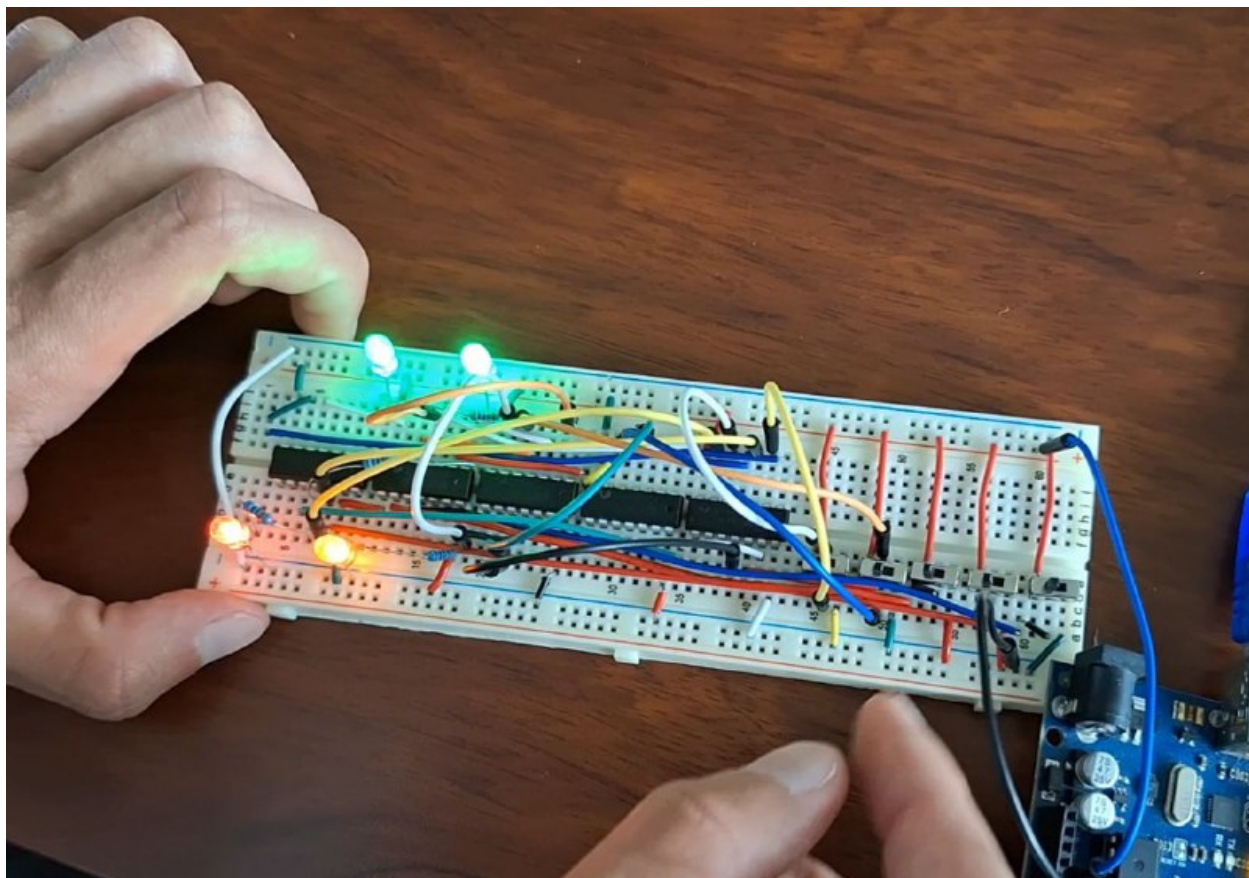


Figura 2.5: Decodificadores en Protoboard

El circuito presente en 2.6 fue el utilizado para el taller "Diseño de un decodificador". De este circuito se partió para el resto de la implementación del proyecto. Siendo necesario sustituir los switch de entrada por el arreglo de LDR's. Así como modificar el funcionamiento del acumulado, al dejar de lado los switches del acumulado para darle lugar al registro. Además queda pendiente la implementación del motor así como el botón accionador.

## 2.6. Diseño circuito LDR: 7/09/2024

Para el diseño del circuito con LDR, se replicó un esquema básico para encender un LED al detectar una sombra sobre la fotoresistencia. Este circuito fue replicado para todo el arreglo de LDR.

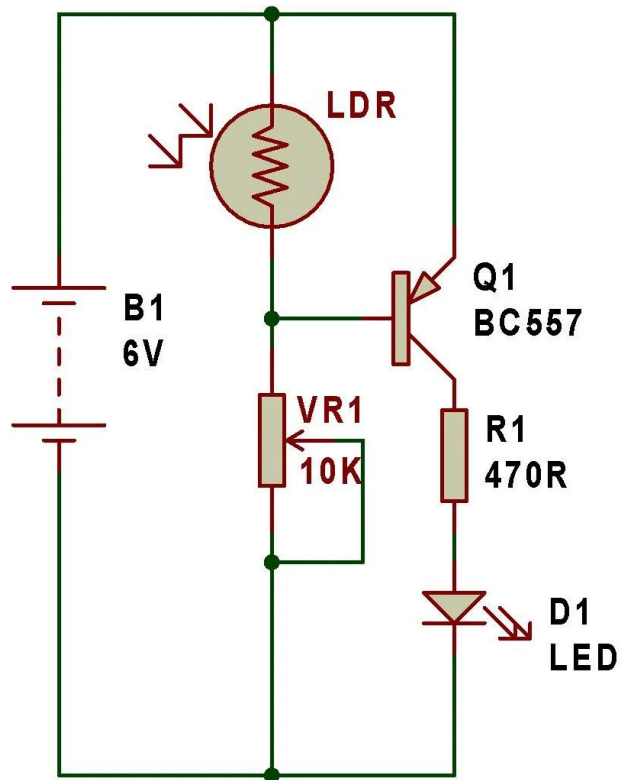


Figura 2.6: Diagrama activación de un LED haciendo uso de un LDR

Se tomaron los voltajes de  $V_{R1}$  de los primeros cuatro segmentos del LDR, los cuales se suministraron a una compuerta NOT para generar las señales binarias que serían procesadas por el primer decodificador.

En esta etapa se presentó un pequeño desafío, puesto que no todos los leds encendían al colocar la mano sobre la fotoresistencia. Se logró solucionar al reemplazar resistencias que se encontraban dañadas.

## 2.7. 9/09/2024

Se sustituyeron los switches del acumulado, haciendo uso del registro y del botón accionador. Así mismo se sustituyeron los LED's encargados de representar el resultado de la suma por el BCD y el siete segmentos.

## 2.8. 10/09/2024

Para la implementación del motor se decidió que el rango de valores para encender el componente sería cuando la suma arroje un resultado que sea 2 o 3. Esto se logra de manera muy sencilla ya que solamente se debe conectar un único cable al bit más significativo que se le suministra al BCD, siendo simplemente la conexión de un cable. Esto se representa a través de la siguiente tabla

Cuadro 7: Karnaugh para el  $Z_1$  del deco de Dedos

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	1

Teniendo que  $Y = A$

Además, fue necesario aplicar un acople para el motor para que no consumiera una corriente elevada, se decidió acudir al siguiente diagrama:

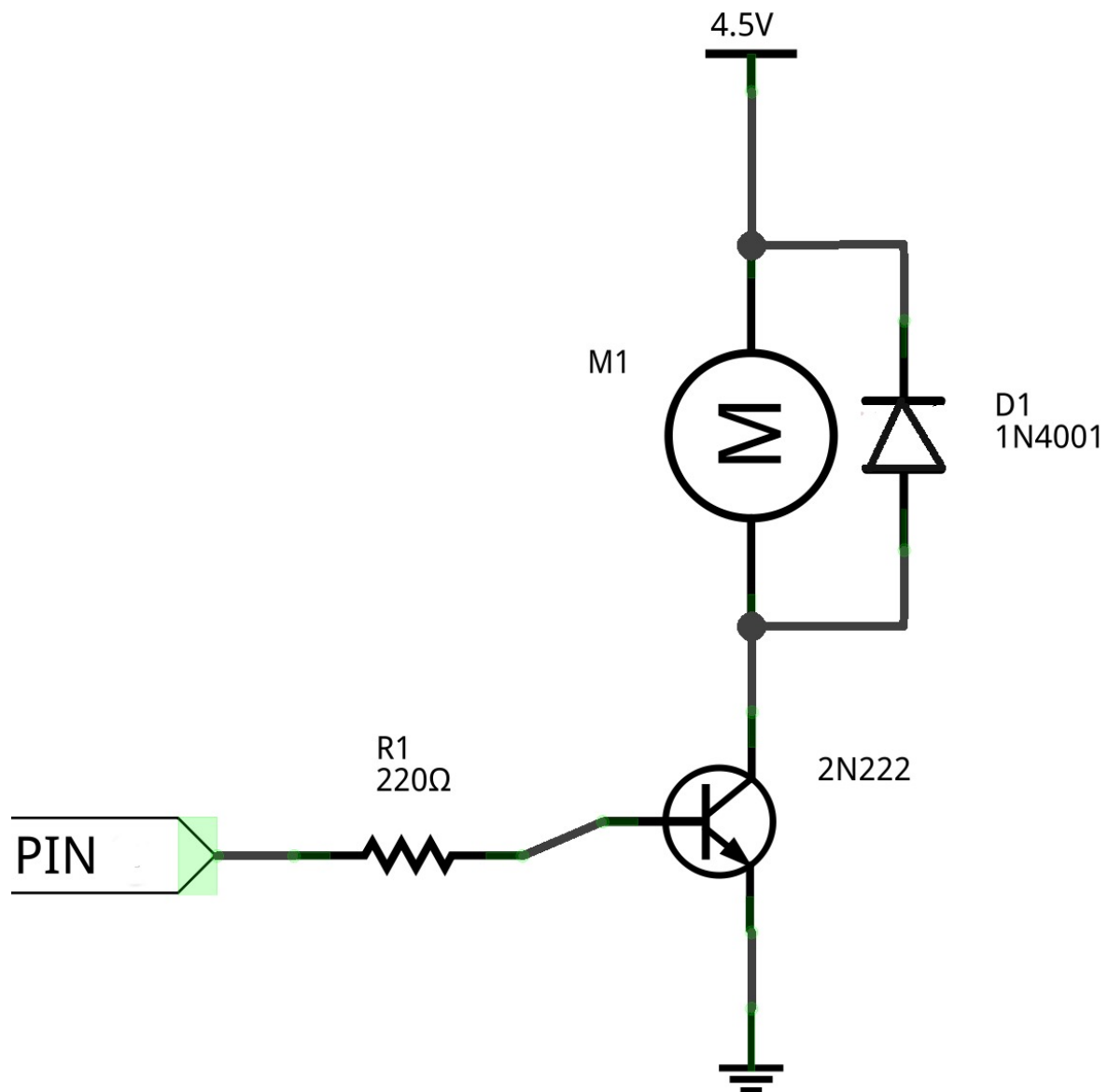


Figura 2.7: Diagrama del acople del Motor