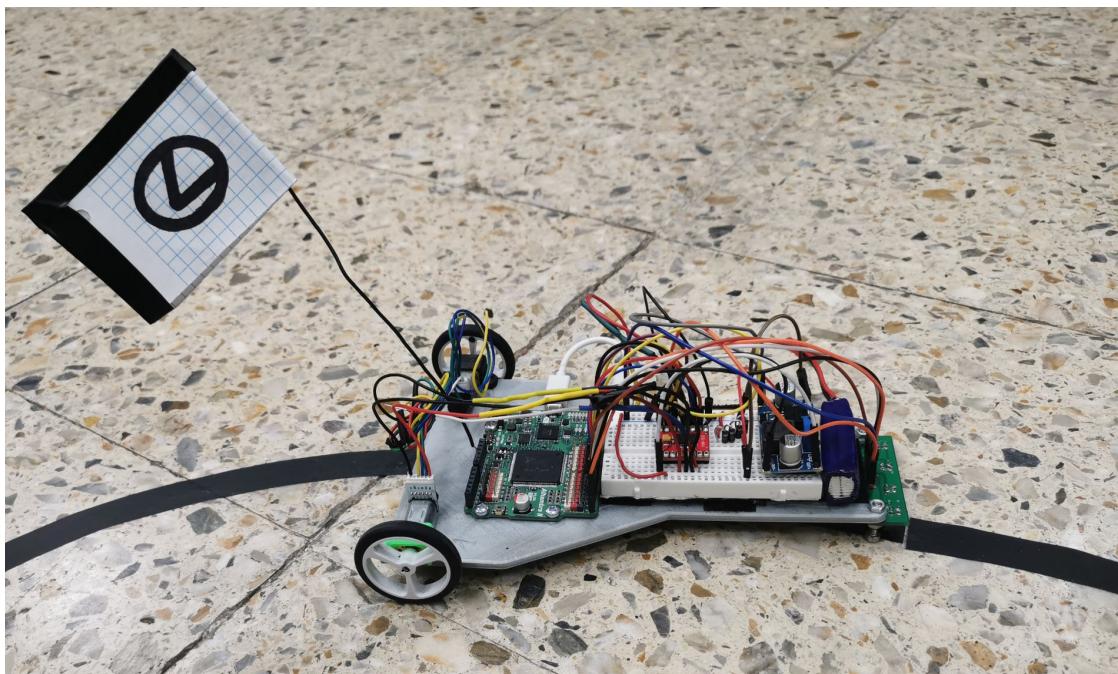


GUIA

COCHE SIGUE LINEAS



NIKOLAY FERNADEZ
RUBEN LOZANO
MARTIN VERDUGO

ÍNDICE

1. INTRODUCCIÓN.....	3
2. LISTADO DE MATERIALES	3
2.2. Protoboard.....	4
2.3. Sensores infrarrojos CNY70.....	4
2.4. Motores DC: 6V 1500 rpm (10:1).....	5
2.5. Batería de litio 0.3A y 7.4v.....	5
2.6. Controlador de motor TB6612FNG	6
2.7. Fuente conmutada DC-DC LM2596.....	6
2.8 Alhambra II - FPGA	6
3. MONTAJE.....	7
4. PROGRAMACIÓN	11
4.1 TABLA DE LA VERDAD.....	11
4.2 PROGRAMA.....	12
5. BIBLIOGRAFÍA.....	15

1. INTRODUCCIÓN

A través de este documento podrás llevar a cabo el completo montaje y programación paso a paso de nuestro coche sigue-líneas sin necesidad de buscar información en internet.

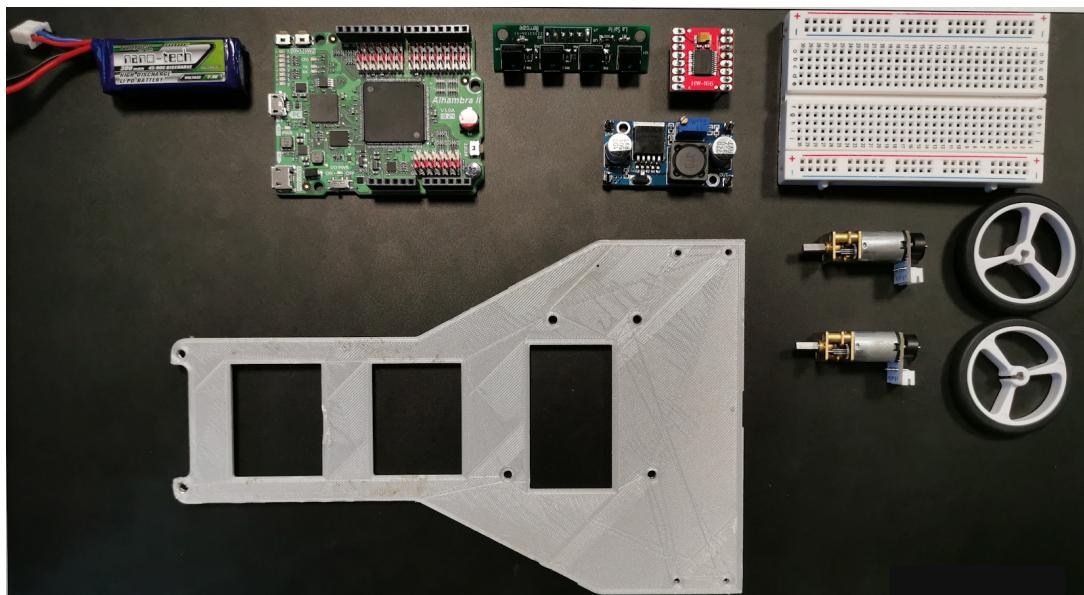
Para ello vamos a estructurar el documento en diferentes apartados:

- LISTADO DE MATERIALES
- MONTAJE
- ESQUEMAS
- PROGRAMACIÓN

Una vez explicados todos los puntos veremos el resultado final, tanto a través de videos como por imágenes

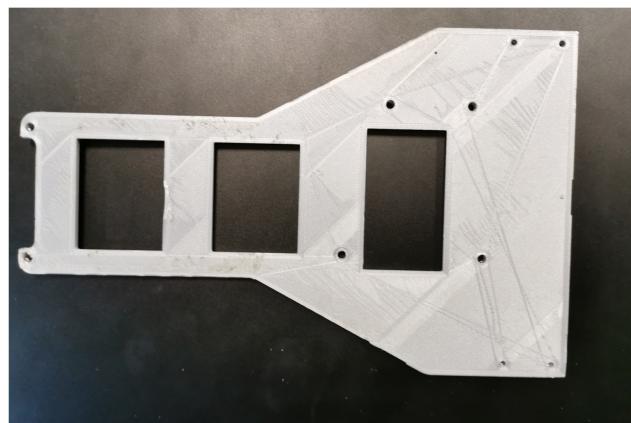
2. LISTADO DE MATERIALES

Para llevara cabo el coche hemos utilizado todos los componentes que puedes observar en la siguiente foto y a continuación los comentaremos uno por uno:



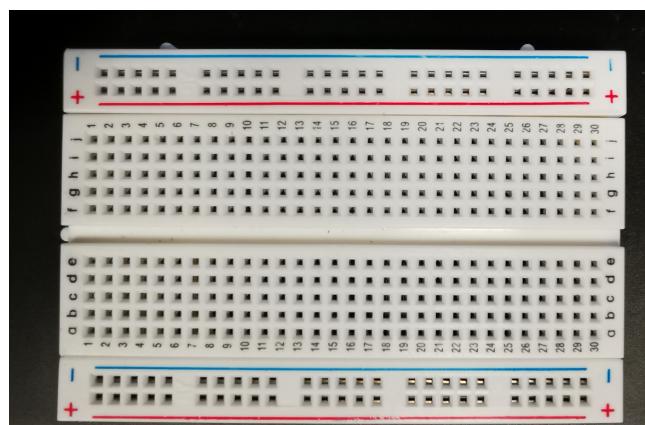
2.1. Chasis:

El chasis de nuestro coche está creado a través de impresión 3D. Es un chasis genérico para todos los grupos.



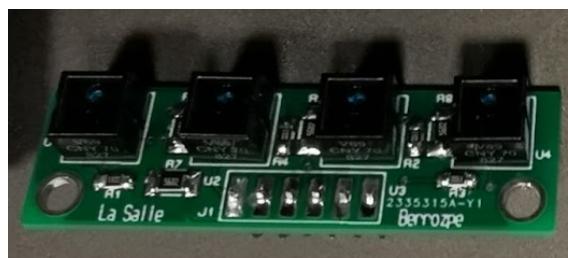
2.2. Protoboard:

La protoboard la utilizamos para facilitar las conexiones entre los elementos a través de conectores macho-hembra.



2.3. Sensores infrarrojos CNY70

Estos sensores infrarrojos detectarán la línea que debe de seguir el coche. Para ello disponen de un emisor LED y un receptor los cuales emitirán y recibirán de vuelta un reflejo de luz detectando el color negro de la linea o nada en el caso de que no se refleje la luz en ningun lugar.



2.4. Motores DC: 6V 1500 rpm (10:1)

Hemos utilizado dos motores DC de 6V que están conectados a un controlador y gracias a ellos haremos girar las ruedas a la velocidad y en sentido de giro que queramos.



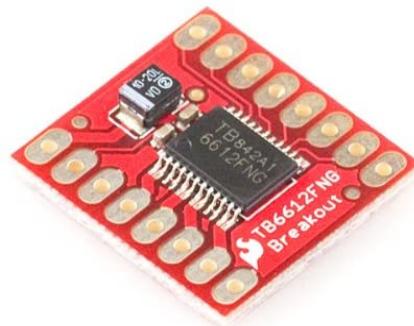
2.5. Batería de litio 0.3A y 7.4v

Gracias a esta hemos alimentado por un lado la fuente conmutada y la placa FPGA. Es una batería de litio de 0.3A y de 7.4v de voltaje y tiene unas dimensiones: 44 x 17 x 12 mm.



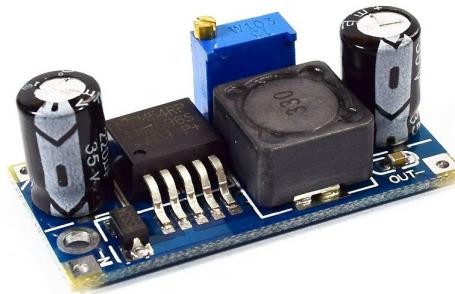
2.6. Controlador de motor TB6612FNG:

Este módulo es indispensable para poder manejar dos motores de corriente continua. Variando tanto la velocidad como el sentido de giro.



2.7. Fuente conmutada DC-DC LM2596

Esta fuente conmutada contiene un regulador que nos permite obtener un voltaje regulado a partir de una fuente de mayor voltaje. La cual nos va a permitir regular de 7,4V de la batería a 6V para así alimentar los motores con una alta eficiencia.



2.8 Alhambra II - FPGA

Es una placa de hardware libre compatible con Ice Studio en el cual hemos creado nuestro circuito para el correcto funcionamiento de nuestro coche.



3. MONTAJE

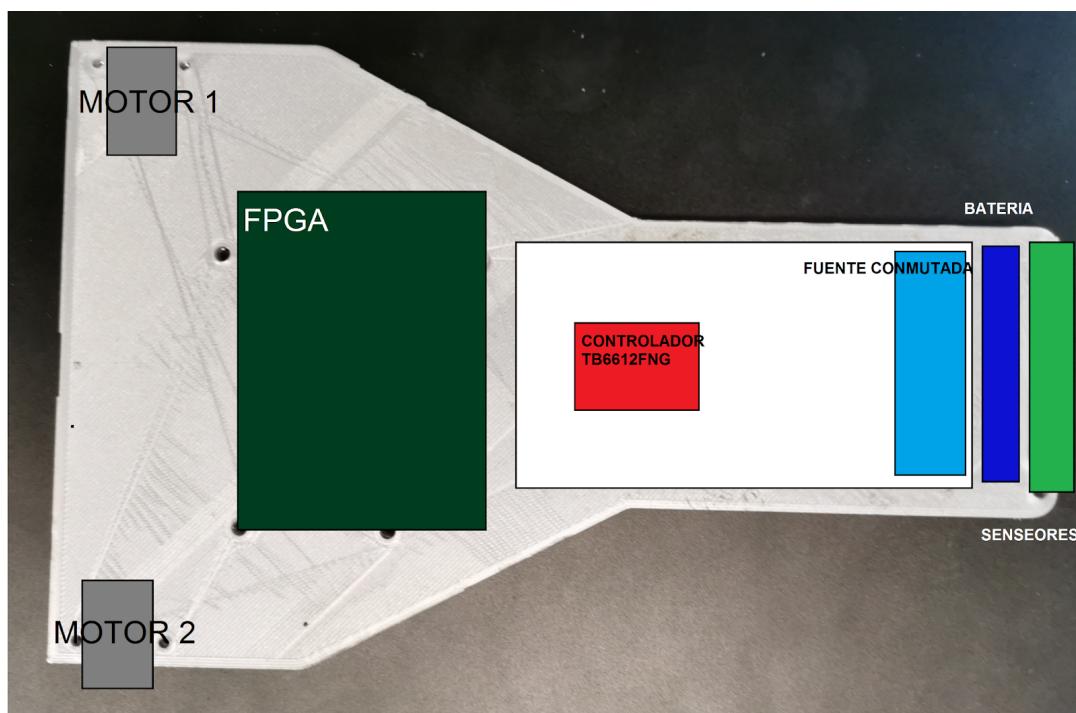
Una vez que tenemos claros todos los componentes necesarios para crear nuestro coche lo primero que tenemos que hacer es comprobar que todos los materiales hacen su correcta función.

En caso de no saber cómo funcionan, debemos leer sus respectivos datasheets y buscar información al respecto. Nosotros verificamos que todos los componentes funcionan correctamente por separado para luego no tener ningún problema.

Una vez realizado todo ello, empezaremos a montar el coche paso a paso.

PASO 1

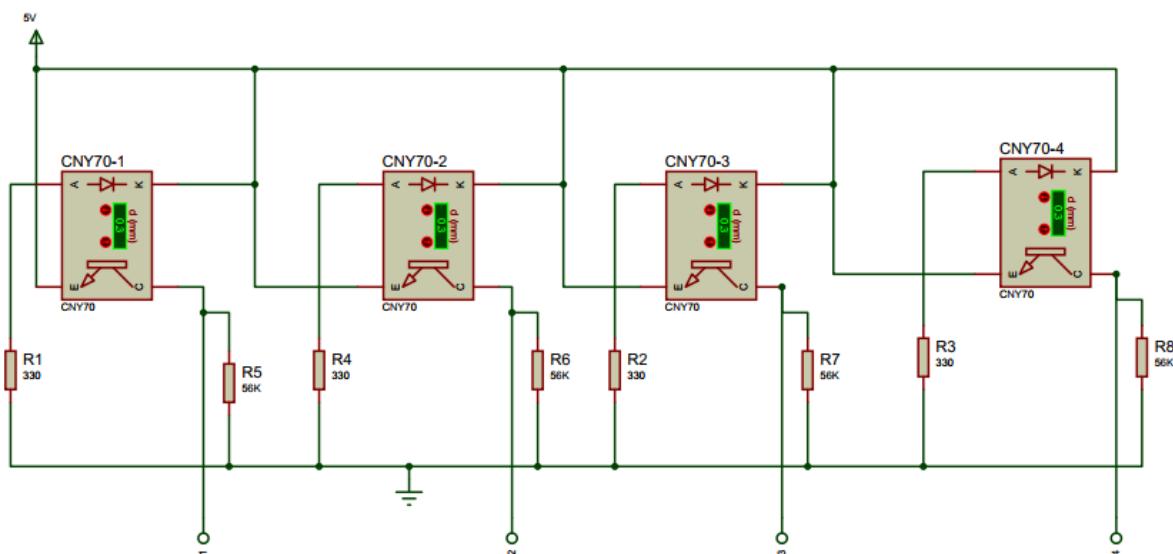
Primero de todo, haremos un croquis de cómo situar todos los elementos y sus respectivos espacios en el chasis.



En nuestro caso lo primero que montamos fueron los motores. Una vez que verificamos que los dos motores funcionaban correctamente los colocamos con sus correspondientes tornillos y tuercas amarrados a la superficie del chasis para que no se moviesen y funcionasen correctamente.

PASO 2

En segundo lugar, probamos que los sensores CNY70 funcionaban correctamente y para ello tuvimos que estañar unos pines tanto para la entrada de voltaje y tierra como para cada uno de ellos, 6 pines en total. Seguido os adjunto el esquema que diseñamos e hicimos para poder ver que los 4 sensores funcionaban correctamente.



Los sensores infrarrojos tienen un emisor y un receptor y por lo tanto cuando emite la luz y debajo de este colocamos algo negro como una cinta, la luz se refleja y el receptor recibe la señal significando que el sensor detecta correctamente.

PASO 3

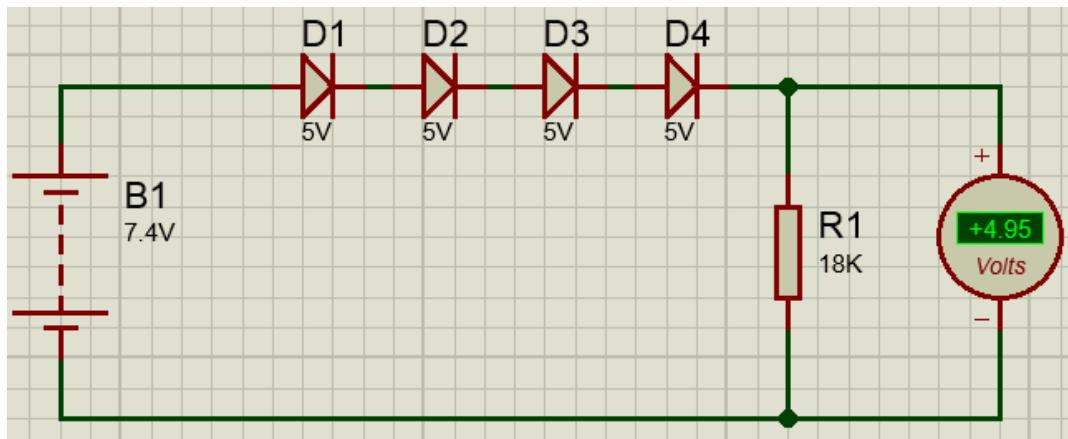
En el tercer paso lo que hicimos fue regular la fuente conmutada y así poder conseguir los 6V para alimentar los motores posteriormente.

Para ello, conectamos a un borne los 7,4V que nos daban la batería y haciendo pruebas con el polímetro fuimos regulandolo conseguimos tener 6V a la salida.

Estas fuentes conmutadas son fuentes de alimentación reducidas que convierten de una fuente de alimentación de mayor voltaje a un voltaje menor para ello utilizan unos transistores mosfets que no dejan pasar la corriente y unos condensadores para así poder almacenar la corriente y más tarde emitirla.

PASO 4

En este paso lo que haremos es convertir 7,4V de la batería en 5V para así poder alimentar la FPGA y el controlador TB6612FNG. Para ello colocaremos 4 diodos en serie para poder conseguir los 5V necesarios. Aquí os adjuntamos la prueba que hicimos en proteus para poder después llevar a cabo la práctica en físico sin tener problemas.



PASO 5

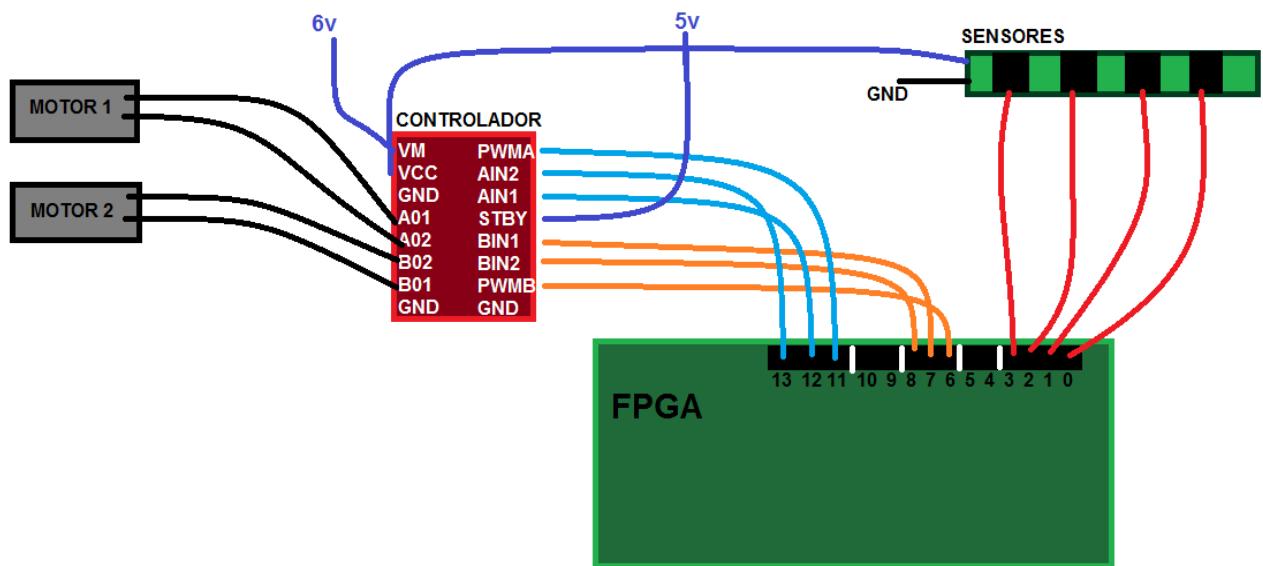
A continuación, podemos observar el esquema general de cómo hemos conectado todos los componentes para llevar a cabo nuestro coche.

Los motores los conectamos de tal forma que el motor 1 estará conectado al A01 y A02 mientras que el motor dos estará conectado a B01 y B02.

Por otra parte, al controlador llegarán 5V para alimentarlo y conectarlos con el STBY (porque sino no funciona) y también llega a los 6V necesarios para alimentar los motores.

En el caso de la FPGA utilizaremos diferentes pines, los 4 primeros del 0 al 3 estarán conectados a los diferentes sensores.

Los siguientes que utilizaremos serán los numerados del 6 al 8 los cuales son **PWM** **B** (controlaremos las distintas velocidades del motor 2) / **BIN1** (Girar hacia delante) / **BNI2** (girar hacia atrás). Los pines del 11 al 13 los utilizamos de la misma manera pero para el motor 1



4. PROGRAMACIÓN

Para poder llevar a cabo el correcto funcionamiento de nuestro coche hemos utilizado el programa Icestudio el cual es compatible con Alhambra II y nos permite crear nuestro circuito

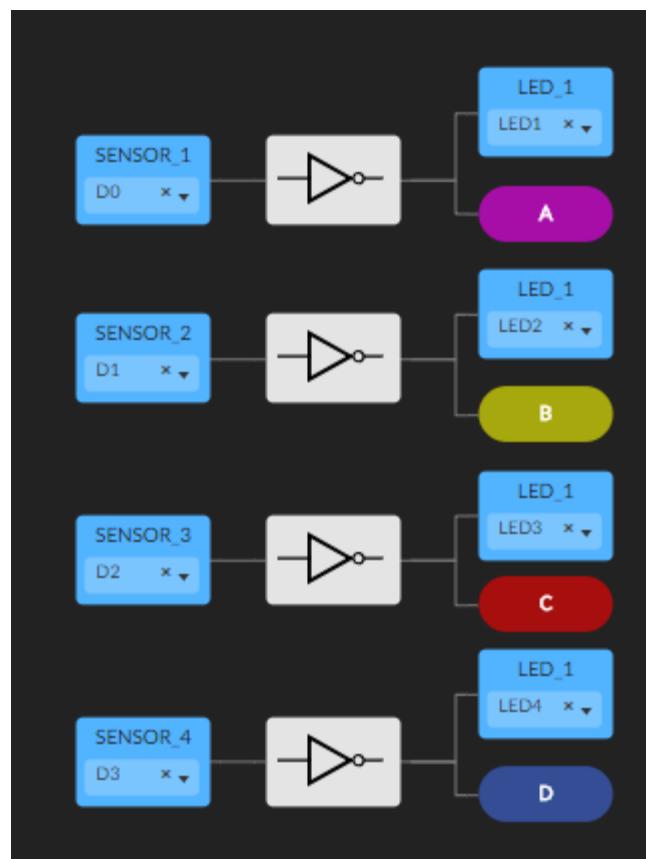
Teniendo en cuenta como queremos que detecten nuestros sensores hemos creado esta tabla de la verdad

4.1 TABLA DE LA VERDAD

A	B	C	D	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	1	1
0	0	1	0	0	1	0	0	0	0	1	0
0	0	1	1	1	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	0	1	0	0
0	1	0	1	x	x	x	x	x	x	x	x
0	1	1	0	0	1	0	0	0	1	0	0
0	1	1	1	0	0	1	0	0	0	1	1
1	0	0	0	1	1	1	1	1	0	0	0
1	0	0	1	x	x	x	x	x	x	x	x
1	0	1	0	x	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x	x
1	1	0	0	0	0	1	1	1	0	0	0
1	1	0	1	x	x	x	x	x	x	x	x
1	1	1	0	0	0	1	1	0	0	1	0
1	1	1	1	0	0	1	1	0	0	1	0

4.2 PROGRAMA

Primero de todo se asignan las entradas de los sensores, en nuestro caso nos daban señales invertidas a las que posteriormente necesitábamos por lo que le pusimos una puerta NOT a cada sensor unida a una etiqueta para poder usarlas después. A su vez, los conectamos a la salida de los leds que incorpora la placa de Alhambra para ver cual se activa en cada momento.



A continuación con los sensores ya invertidos, generamos 4 etiquetas para las 4 situaciones que se nos pueden dar en cada motor, velocidad máxima velocidad media velocidad lenta y giro invertido.

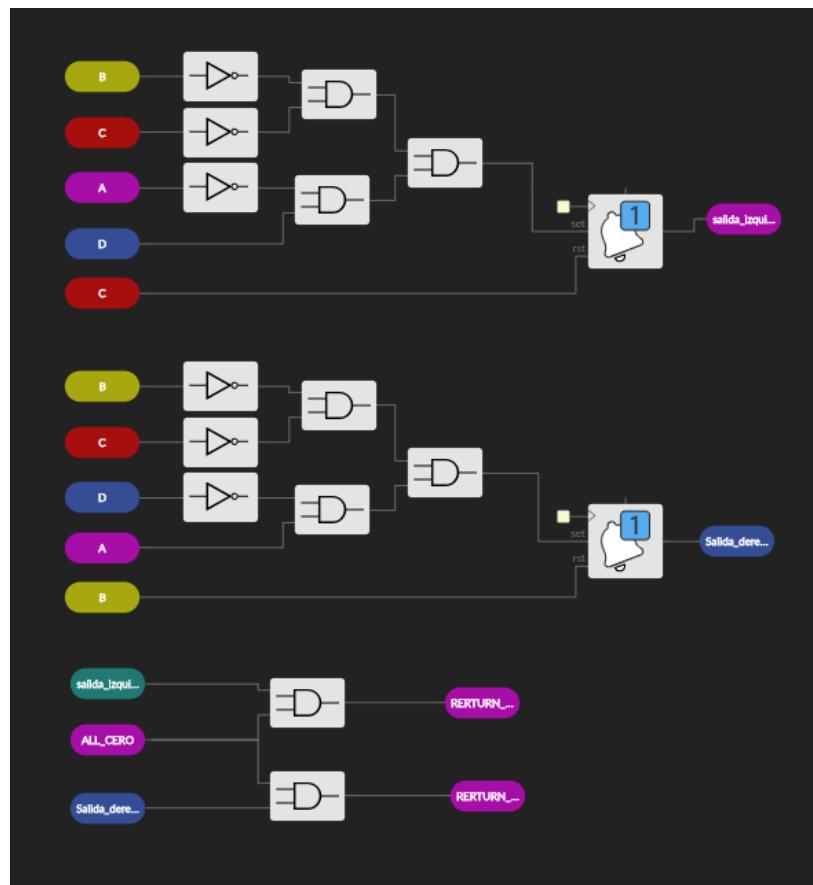


Con las 4 situaciones ya generadas pasamos a definir las velocidades. Para definir las velocidades necesitamos generar una modulación por ancho de pulso (PWM) para la cual nos tuvimos que descargar una colección de internet que incorporaba un bloque nuevo que nos genera una señal PWM. Con el nuevo bloque y las situaciones anteriores definimos las 3 velocidades mediante diferentes valores en un bus de 8 bits que entra al bloque para generar la señal que se conecta directamente a un pin de salida por motor.

Las 2 salidas inferiores son para gestionar la dirección de giro. No es posible que se activen las dos simultáneamente por lo que se ha programado de manera que una anule la contraria.



Para evitar que el coche se salga de la línea y se pierda, hemos programado una memoria que en caso de que el coche se salga de la línea recordará por qué lado se ha salido y se mantendrá girando en esa dirección buscando volver a su línea otra vez.



5. BIBLIOGRAFÍA

Para llevar a cabo nuestro coche sigue líneas hemos buscado información en diferentes páginas webs

Electrónica digital

<http://myelectronic.mipropia.com/digital.html?i=1>

Puertas lógicas:

[https://www.logicbus.com.mx/blog/compuertas-logicas/#:~:text=Las%20Compuertas%20L%C3%B3gicas%20son%20circuitos,binarias%20\(suma%2C%20multiplicaci%C3%B3n\).](https://www.logicbus.com.mx/blog/compuertas-logicas/#:~:text=Las%20Compuertas%20L%C3%B3gicas%20son%20circuitos,binarias%20(suma%2C%20multiplicaci%C3%B3n).)

<https://descubreaduino.com/puertas-logicas/>

Programa coche sigue líneas (copiado)

<http://robots-argentina.com.ar/didactica/circuito-y-detalles-del-proximo-robot-el-sigue-lineas/>

Información sobre FPGA

<http://fpgawars.github.io/>

<https://www.intesc.mx/fpga-que-son-para-que-sirven-y-aplicaciones-2/>

<https://www.hwlibre.com/fpga/>

Motores

<https://www.amazon.es/Cikuso-reduccion-micro-velocidad-engranaje-terminales/dp/B07K4YZFQT?th=1>

Controlador de motores

<https://www.luisllamas.es/arduino-motor-dc-tb6612fng/>

Placa FPGA Alhambra II

<http://ultra-lab.net/producto/alhambra-ii-fpga-board-placa-fpga-abierta-alhambrabits-alhambraii/>

Sensor CNY70

<http://www.digitronik.pe/catalogo/kits-proyectos/sensores-varios/sensor-optico-reflectante-cny70>

Tecnología TTL y CMOS

<https://micros.grup6.com/diferencia-entre-ttl-vs-cmos/>

<https://www.docscity.com/es/ttl-y-cmos-diferencias/5724882/#:~:text=En%20resumen%20podemos%20decir%20que,dise%C3%B1ada%20para%20un%20bajo%20consumo.&text=consiguir%20lo%20mejor%20de%20ambas,consumo%20y%20una%20alta%20velocidad.>