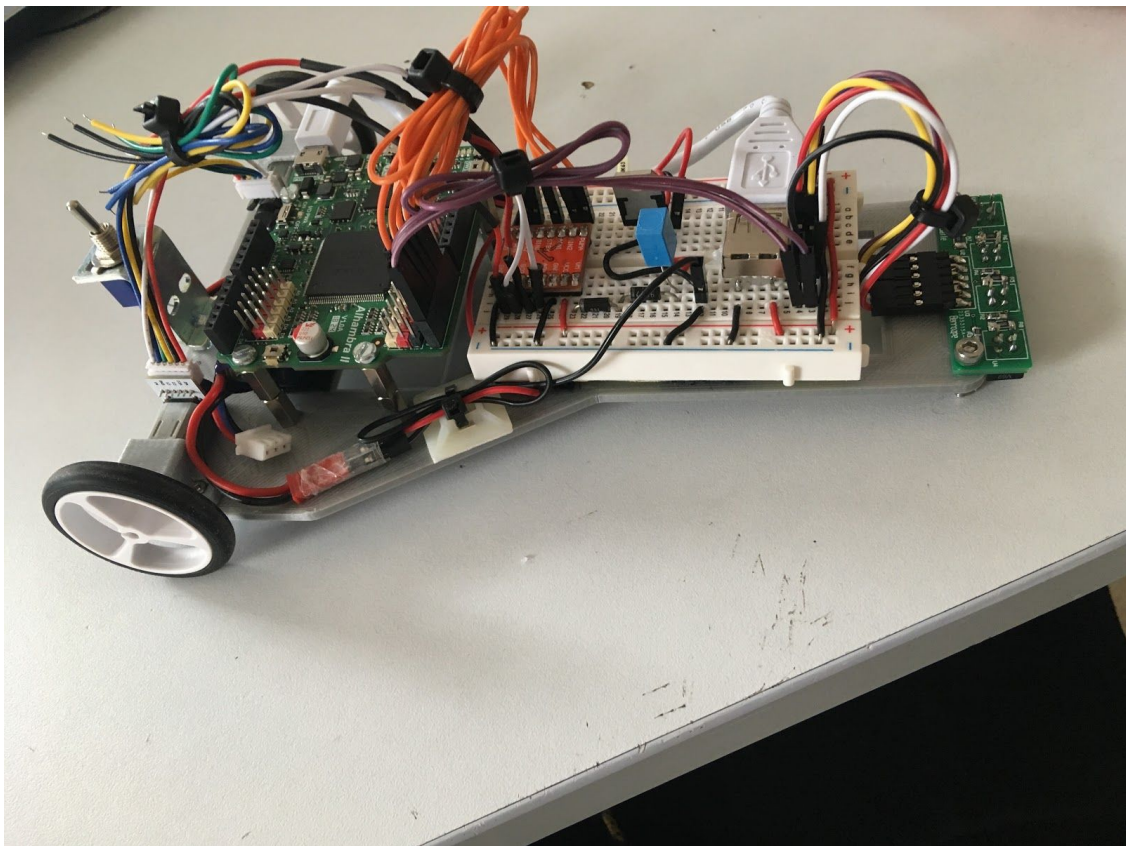


# LERRO JARRAITZAILEA



EKAIN TORRES  
PATRICIA SANZ  
XUBAN MARTIARENA  
G1E

# AURKIBIDEA

<b>LISTA DE MATERIALES:</b>	<b>2</b>
<b>ELEMENTOS PRINCIPALES</b>	<b>2</b>
LM2596 DC-DC	2
TARJETA SENSORES	2
SENSORES CNY70	2
RESISTENCIAS SMD	3
ALHAMBRA II	4
DRIVER TB6612FNG	5
<b>ESQUEMA</b>	<b>6</b>
EXPLICACIÓN circuito	6
<b>TABLAS DE LA VERDAD / KARNAUGH / ECUACIÓN / CIRCUITO CON PUERTAS LÓGICAS.</b>	<b>7</b>
<b>CONFIGURACIÓN DEL HARDWARE</b>	<b>8</b>
<b>MONTAJE:</b>	<b>9</b>
<b>ELECTRONICA DIGITAL</b>	<b>12</b>
AND:	12
OR:	12
NOT:	12
TABLA DE LA VERDAD	13
TABLAS KARNAUGH	14
<b>BIBLIOGRAFIA</b>	<b>15</b>

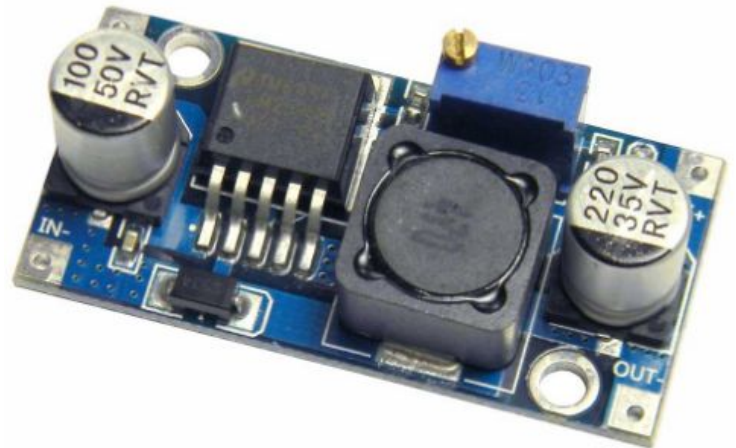
## LISTA DE MATERIALES:

Base coche / chasis (pieza creada con impresora 3D)  
2 Ruedas  
2 Motores  
LM2596 DC-DC  
CNY70 SENSORES  
ALHAMBRA II  
DRIVER TB6612FNG

## ELEMENTOS PRINCIPALES

### LM2596 DC-DC

El convertidor de voltaje DC-DC Step-Down 3A LM2596 su función es entregar un voltaje de salida diferente al de entrada. Soporta corrientes de salida de hasta 3A, voltaje de entrada entre 4.5V a 40V y voltaje de salida entre 1.23V a 37V. El voltaje de salida deseado se selecciona mediante un potenciómetro multivuelta.

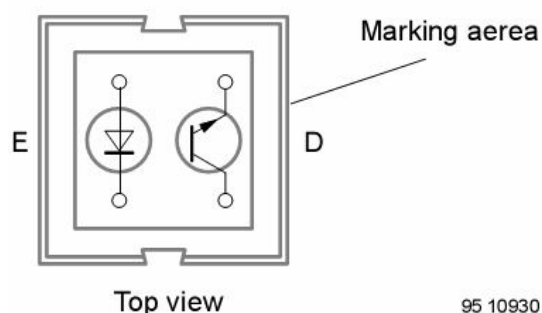


### TARJETA SENSORES

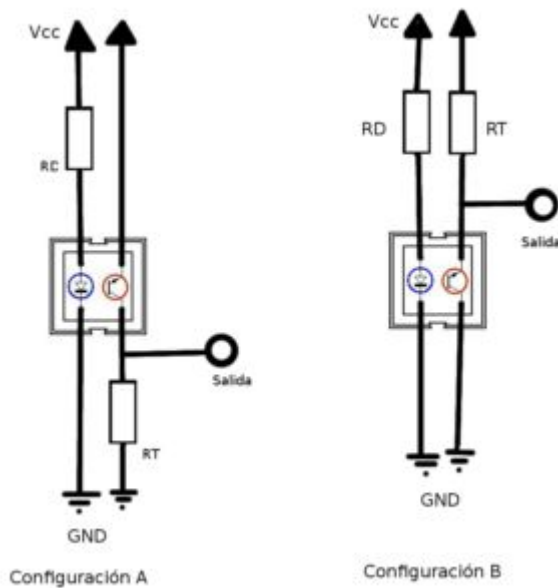
#### SENSORES CNY70

El sensor CNY70 está compuesto por dos elementos, un led infrarrojo y un receptor capaz de ver en este espectro de luz.

Cuando el LED infrarrojo del CNY70 es alimentado comienza a emitir luz en un espectro que es imperceptible al ojo humano. Este haz de luz, si se encuentra con un objeto a poca distancia, es capaz de rebotar e incidir sobre el receptor del sensor.



Dependiendo de cómo hayamos montado el sensor, el receptor entregará un 1 o un 0 cuando la luz le es devuelta y el valor contrario en caso de estar «a oscuras».



Estos son las dos configuraciones del CNY70. En la A, el emisor devolverá un 1 cuando esté recibiendo luz y un 0 en caso contrario. En el modo B el emisor devolverá un 1 si no recibe nada y un 0 en caso de estar recibiendo luz.

### RESISTENCIAS SMD

Hemos utilizado dos resistencias, las dos de tipo SMD.

Para los Led hemos utilizado resistencias con valor 5602 ( $56K = 56000 \Omega$ )

Para los sensores hemos utilizado resistencias con valor 3300 ( $0,33K = 330 \Omega$ )

Para saber el valor resistivo de las resistencias SMD el cálculo debe realizarse así.

## Resistencias SMD con código de 3 dígitos



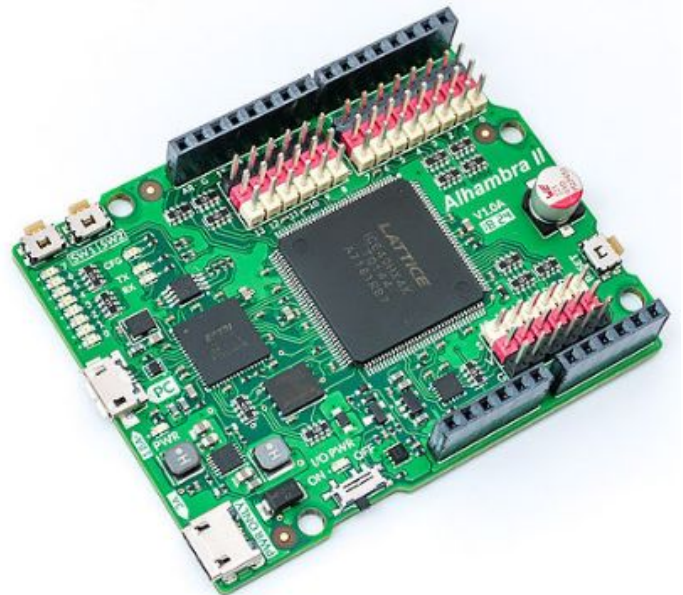
Resistencias SMD de precisión  
con código de 4 cifras.  
Valores iguales o mayores de  $100\ \Omega$



## ALHAMBRA II

FPGA son las siglas de Field Programmable Gate Array, las placas FPGAs son dispositivos digitales que son capaces de configurarse prácticamente para cualquier aplicación, son capaces de trabajar muchos procesos en paralelo esto quiere decir que todo lo que esté programado se realizará a la vez y no de manera secuencial como hemos visto en Arduino por ejemplo.

Lo interesante está en que se pueden “programar” las conexiones entre las compuertas y FLIP-FLOPs de modo que se puede crear cualquier dispositivo digital



que podamos imaginar, los únicos limitantes son la frecuencia y la cantidad de compuertas.

En nuestro caso hemos utilizado es la FPGA Alhambra II ya que como citan sus propios creadores es de código abierto:

*“Esta placa trata sobre la exploración del lado de código abierto de los FPGA . Sabemos que hay FPGA más potentes. Sabemos que son herramientas de software muy increíbles que pueden hacer muchas cosas ... pero no son de código abierto. Entonces, si te gusta la libertad también, esta tabla es para ti.”*

## DRIVER TB6612FNG

Este elemento es un controlador de motores que nos permite manejar dos motores de corriente continua desde nuestra Alhambra II, variando tanto la velocidad como el sentido de giro.

El TB6612FNG puede ser considerado una versión mejorada del L298N.(controlador que sopesamos poner en un principio). Al igual que esté, internamente está formado por dos puentes-H, junto con la electrónica necesaria para simplificar su uso, y eliminar posibles cortocircuitos por errores de uso.

Sin embargo, en el caso del TB6612FNG los puentes-H están formados por transistores MOSFET, en lugar de transistores BJT como en el L298N. Esto permite que el TB6612FNG tenga mejor eficiencia y menores dimensiones que el L298N.

El TB6612FNG también permite controlar intensidades de corriente superiores, siendo capaz de suministrar 1.2A por canal de forma continua, y 3.2A de pico.

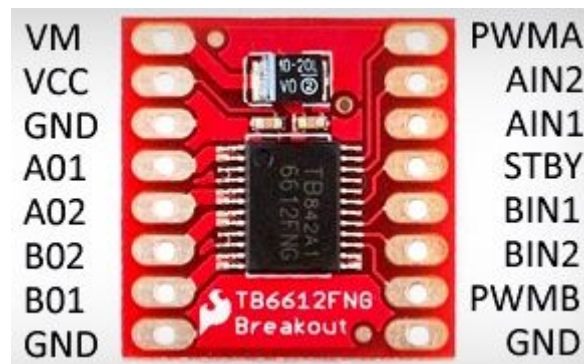
El TB6612FNG dispone de dos canales, por lo que es posible controlar dos motores de corriente continua de forma independiente.

En cada canal podemos controlar el sentido de giro y la velocidad, para lo cual admite una señal PWM de frecuencia máxima de 100 kHz

El TB6612FNG dispone de protecciones térmicas, de inversión de corriente en la fuente de suministro de los motores, condensadores de filtrado en ambas líneas de alimentación, detección de bajo voltaje, y protecciones contra las corrientes inducidas en los motores.

El TB6612FNG también incorpora un modo de Standby, que desactiva por completo el controlador, entrando en un modo de ahorro de energía.





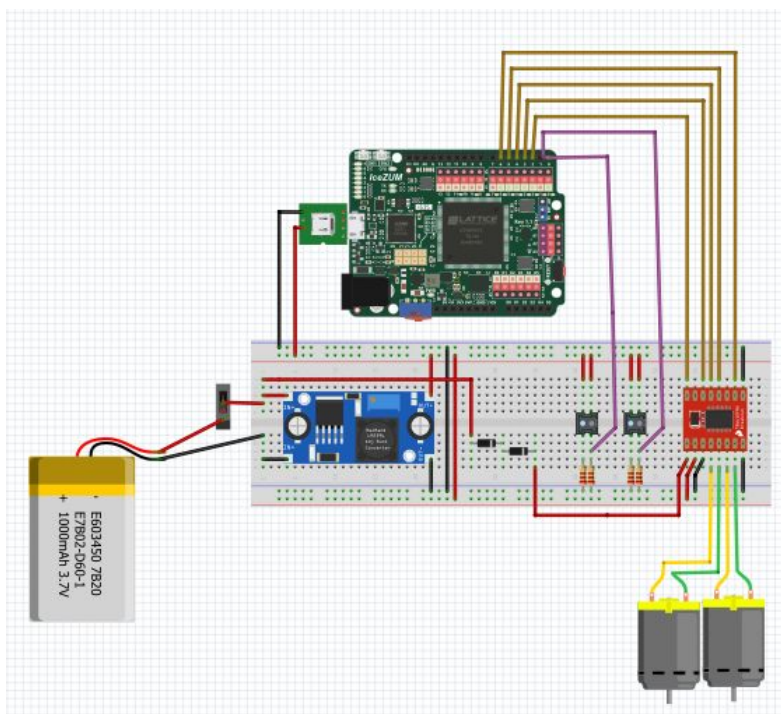
El esquema de montaje no es demasiado complicado. Por un lado, suministramos la tensión que alimentará el motor desde una fuente de alimentación externa, mediante el pin VM. La tensión máxima es de 15V.

Además, tenemos que alimentar la electrónica del módulo mediante el pin VCC. El rango de tensión para VCC es 2.7 a 5.5V.

Para el control del módulo los pines AIN1, AIN2 Y PWMA controlan el canal A, mientras que los pines BIN1, BIN2, y PWMB controlan el canal B.

Finalmente, el pin STBY controla el modo Standby. Debemos ponerlo en HIGH para activar el motor. Podemos conectarlo a un pin digital de Arduino, si queremos poder activar el modo Standby, o conectarlo a VCC si queremos dejarlo permanentemente desconectado.

## ESQUEMA



## EXPLICACIÓN circuito

Vamos a alimentar el circuito con una batería LIPO de 7,4V. Después de la batería, hemos puesto un interruptor para proteger la batería en caso de algún tipo de fallo. Para convertir los 7,4V a 5V hemos puesto un convertidor DC/DC LM2596, y con este vamos a alimentar la FPGA Alhambra II, los sensores

CNY60, y el driver para controlar los motores TB6612FNG.

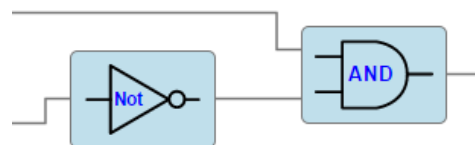
Para alimentar los motores a 6V, hemos puesto dos diodos en serie, para que se queden con 1,4V y tener 6V en los motores.

## TABLAS DE LA VERDAD / KARNAUGH / ECUACIÓN / CIRCUITO CON PUERTAS LÓGICAS.

SS1 Izq	SS2 Dch	PWMA Izq	AIN1 Izq	STBY	BIN1 Dch	PWMB Dch
0	0	0	0	0	0	0
0	1	0	1	1	0	1
1	0	1	0	1	1	0
1	1	0	1	1	1	0

	SS2	
SS1	0	1
0	0	0
1	1	0

$$PWMA = SS1 \cdot \overline{SS2}$$



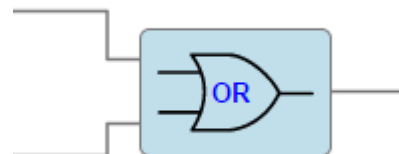
	SS2	
SS1	0	1
0	0	1
1	0	1

$$AIN1 = SS2$$



	SS2	
SS1	0	1
0	0	1
1	1	1

$$STBY = SS1 + SS2$$



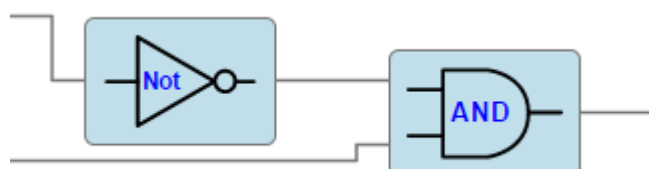
	SS2	
SS1	0	1
0	0	0
1	1	1

$$BIN1 = SS1$$



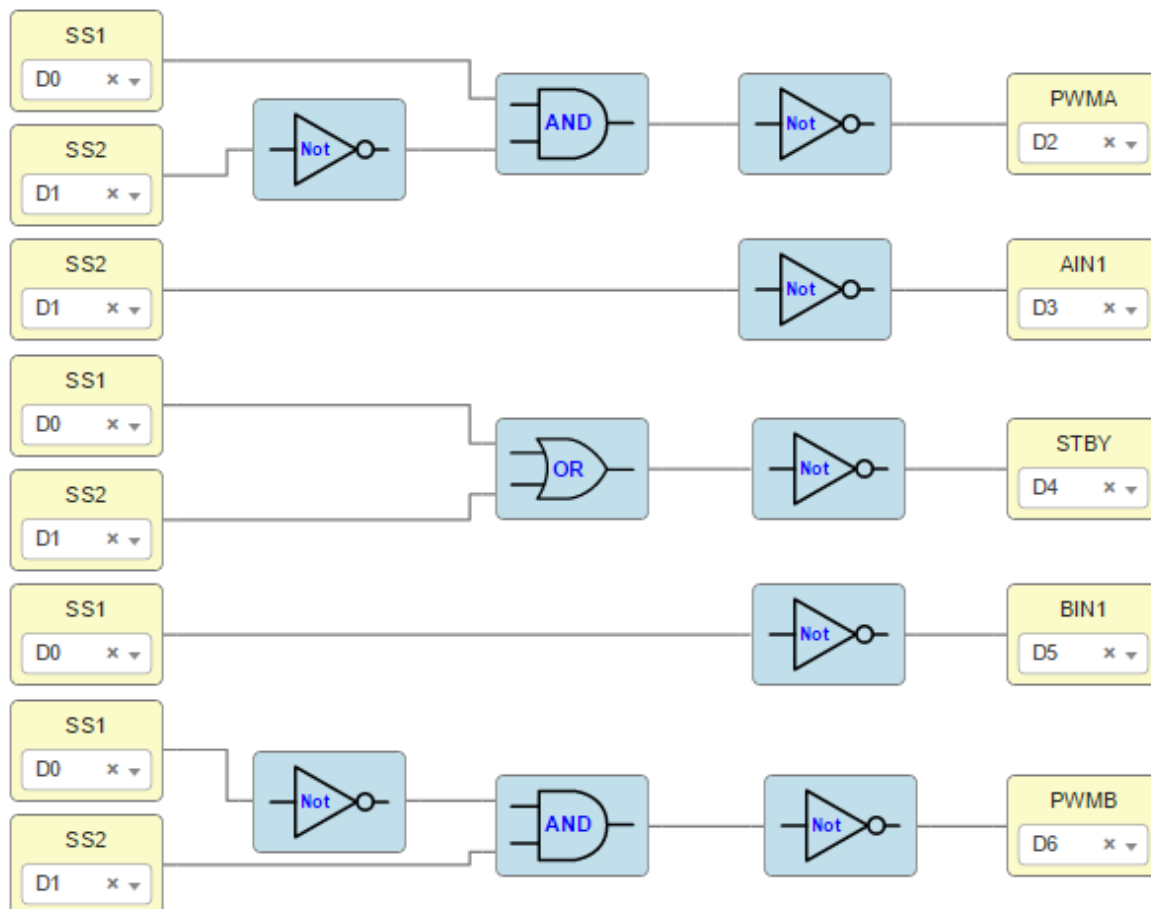
	SS2	
SS1	0	1
0	0	1
1	0	0

$$PWMB = \overline{SS1} \cdot SS2$$





## CONFIGURACIÓN DEL HARDWARE



Hemen insertatu duguna, gure hardwareko konfigurazioa dugu, izan ere, alhambra II, arduinoa ez bezala, FPGA-z baliatzen da eta ez mikroprozesadore batetaz.

FPGA-n programa egin beharrea hardware-a konfiguratzeko dugu, izan ere, honen barrenean hainbat ate logiko daude eta hauen artean bideak sortzen hardware-a konfiguratzeko lor dezakegu.

Gainera, mikroprozesadore bat baino askoz azkarragoa da, izan ere, azken honetan goitik behera irakurtzen joaten da eta baldintzak betetzen direnaren edo ez direnaren harabera akzio bat edo bestea egiten du. Aldiz, ate logikoak erabiliz, bide bat egiten duzu, hortaz, bide hori egingo du zuzenean eta ez da egongo aukera guztiak begiratu arte itxaroten.

Gure konfigurazioan, bi detektorez baliatu gara informazioa biltzeko: SS1(1) eta SS2(0) daudenean, PWMA aktibatzeko esan diogu, izan ere, oso eskuinean dagoela detektatu du eta ezkerreko motorrari polikiago joateko esaten diogu errekifika dezan.

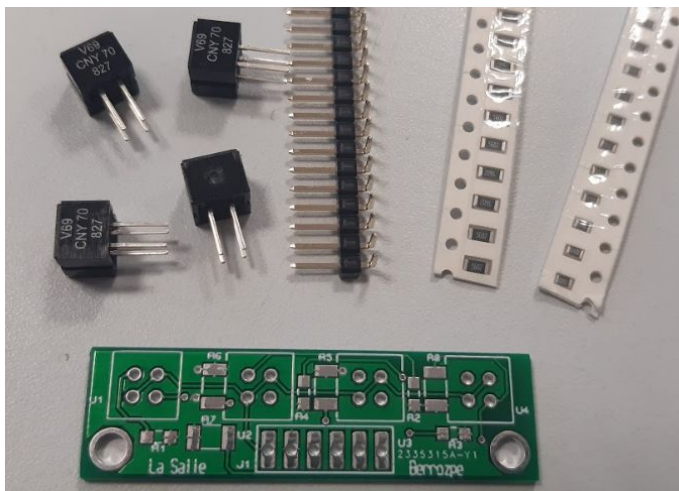
Kasu bera alderantziz gertatuko balitz, hau da, SS1(0) eta SS2(1), PWMB aktibatuko genuke, oraingo honetan, eskuineko motorra izan dadin mantsoago joango dena. STBY, motorrak martxan jartzeko dela jakinik, sensore bat edo bestea detektatzen dagoen bitartean aktibatuta mantentzeko, OR ate logiko batez baliatu gara.

azkenik, SS1(1) egonik, AIN1 aktibatzea jarri dugu eta SS2(1) egotean BIN1. Hauek, motorraren norantza definitzen dutelako.

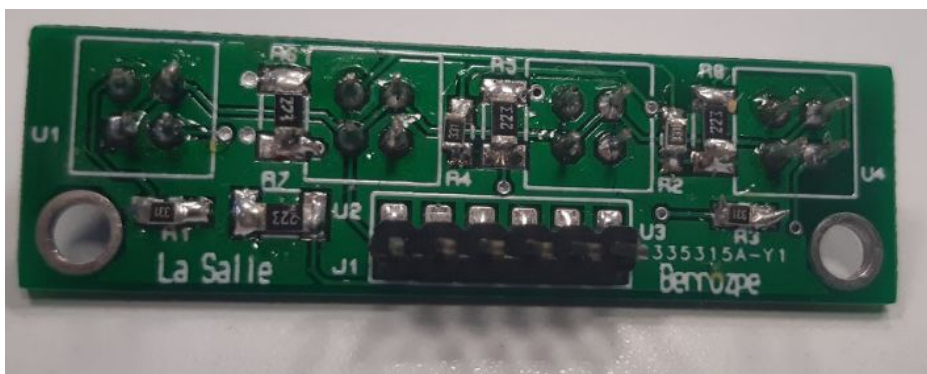
Gainera, ohartuko zinezten bezala, lerro bakoitzaren amaieran NOT bat jarri dugu, 0 bat ematen zigulako eta guk 1 behar genuelako.

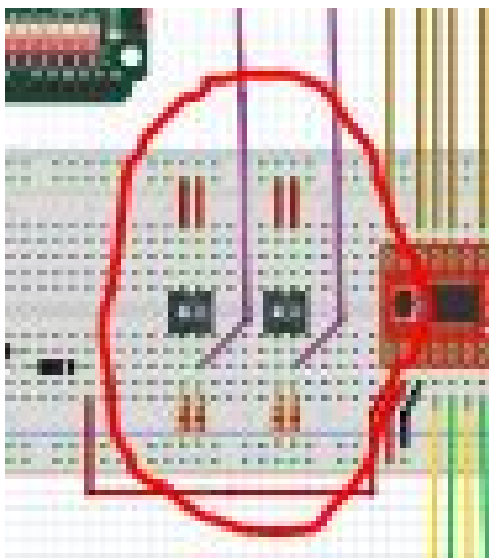
## MONTAJE:

Estos son los componentes citados anteriormente que usaremos para hacer nuestra placa de sensores.



Y esta es nuestra placa de sensores con los elementos ya soldados.



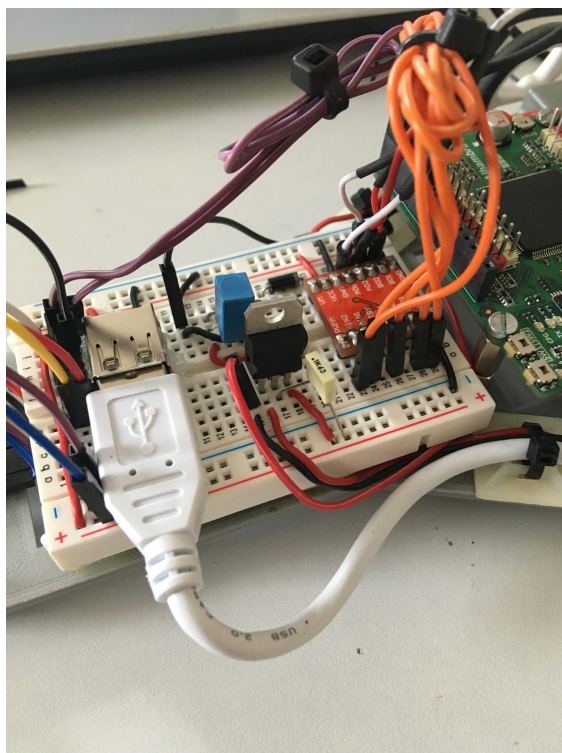


Goiko txartelean eginiko zirkuitua hauxe izango litzateke. Gure kasuan, helburua kotxe bat egitea denez, eta lerro bat jarraitu, zirkuitu inpreso batean egin genuen, kotxearen aurrealdean jarri eta kableak aurrezteko, board-ean lekua aurrezteaz gain.



Konexio guztiak burutu ahal izateko, board bat erabili dugu. Bertan driverra eta kableatu guztiak ipini ditugu.

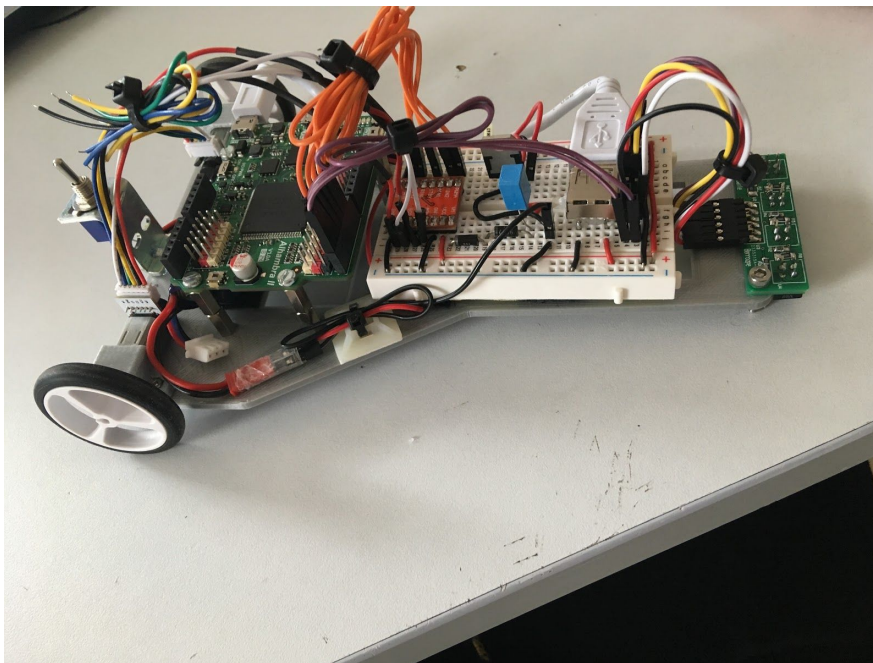
Gainera, hasierako ideia, LM2596 DC-DC konbertidorea erabiliz, alhambra, sensore eta motorrerako driverrean genituen kontrolatuko pinetarako 5V lortzea zen baina geizki funtzionatzen zuenez, 7805 reguladorea erabili behar izan ditugu. Hortaz, board-ean txertatu dugu hemen beheko irudian ikus dezakezuen bezala.



Azkenik, driver eta sensoreetako kableak alhambrara konektatu ditugu, hau baita zirkuituan aginpea edukiko duen elementua.

guztia alimentatzeko, alhambraren azpian ezkutaturik daukagun 3,7Vko bi lipo bateria ditugu lan hau burutzeko, hau da, 7,4V edukiko ditugu. Kontuan harturik motorrak 6Vtara alimentatzen direla, bi diodo jarri ditugu 1,4V horiek jan ditzaten.

Eta nola ez, autoa piztu eta itzaltzeko aukera eduki dezagun, atzekaldean ikusten duzuen interruptorea instalatu diogu.



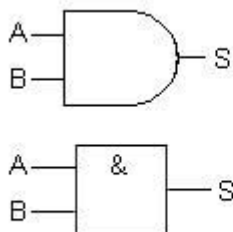
Gure lerro jarraitzaileko kotxearen zatiak osatuz, eta bidean izan ditugun aldaketekin hobekuntzak egin ondoren, ezkerretara ikusten duzue talde lanean garatzea lortu dugun proiektua.

# ELECTRONICA DIGITAL

Proiektua gauzatu ahal izateko, elektronika digitala erabili dugu, hau da ate logikoak baliatu ditugu hardwarea konfiguratzeko.

Erabilitako ate logikoak, AND, OR eta NOT izan dira.

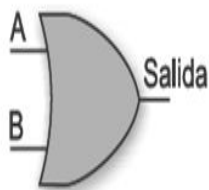
AND:



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

Ezkerreko sinboloa AND ate logikoa da. Honek, dituen sarrera guztietan 1 lortzen badu, irteeran ere 1 emango du, bestela, 0 bat edukiko du beste edozein kasuetan. Hau ondo uler dezazuen, taula bat txertatu dugu sinboloarekin batera.

OR:



Símbolo compuerta OR

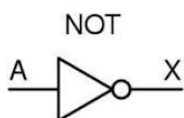
A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	1

Compuerta lógica OR, Tabla de verdad

Honako hau aldiz, Or izeneko ate logiko bat da. Nahiz eta AND sinboloaren antzekoa izan, ez du modu berean funtzionatzen.

Sarreretako batean 1 eduki ezker, irteeran 1 emango du, hau da, nahikoa dauka 1 sarrerako pin bakarrean soilik edukitzea. Hau garbi ikus dezakegu txertatutako taulan.

NOT:



Hirugarren hau izango litzateke ate logiko sinpleena. Izan ere, sarreran 1 baldin badauka, irteeran 0 emango du eta alderantzi izan ezker, hau da, sarreran 0 bat edukiz, irteeran bat emango du. Hortaz, balio baten kontrako balioa lortzeko erabiltzen da.

A	X
0	1
1	0



## TABLA DE LA VERDAD

La tabla de la verdad, es un elemento fundamental a la hora de diseñar circuitos con puertas lógicas, es una tabla en la cual se le da un resultado a las diferentes combinaciones en una función lógica. Por ejemplo, una puerta lógica AND.

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

Digito Decimal	Entradas				Salidas de segmentos						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X

Una tabla de la verdad será más o menos compleja en función de las necesidades del circuito que queramos desarrollar. Primero se sitúan el número de entradas que vayamos a usar, y después a la derecha, se ponen las salidas.

Hecho esto, escribiremos todos los tipos de combinaciones posibles que pueda haber en las entradas, por ejemplo teniendo dos entradas serán 2 elevado al 2 entradas, o sea 4. Siempre será 2 elevado al número de entradas que tengamos, y cada combinación siempre será en código binario, es decir, 0 y 1, es por eso que siempre será un 2 de base.

Al escribir todas las combinaciones de entrada, escribiremos un 0 o un 1 en cada salida, ahí ya viendo lo que queremos que haga cada una de las salidas dependiendo de las combinaciones en las entradas.



## TABLAS KARNAUGH

Es un diagrama utilizado para la simplificación de funciones algebraicas Booleanas.

Las tablas de Karnaugh reducen la necesidad de hacer cálculos extensos para la simplificación de expresiones booleanas, el cerebro humano es capaz de reconocer patrones y otras formas de expresión analítica, permitiendo así identificar y eliminar condiciones muy inmensas.

El mapa de Karnaugh consiste en una representación bidimensional de la tabla de verdad de la función a simplificar. Puesto que la tabla de verdad de una función de  $N$  variables posee  $2^N$  filas, el mapa K correspondiente debe poseer también  $2^N$  cuadrados.

Las variables de la expresión son ordenadas en función de su peso de manera que sólo una de las variables varía entre celdas adyacentes.

La transferencia de los términos de la tabla de verdad al mapa de Karnaugh se realiza de forma directa, albergando un 0 ó un 1, dependiendo del valor que toma la función en cada fila. Las tablas de Karnaugh se pueden fácilmente realizar a mano con funciones de hasta 6 variables.

Obtenemos la tabla de verdad de la función dada y a continuación representamos el mapa de Karnaugh de tres variables

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

AB \ C		0	0	1	1	0	1
		0	1	1	0	1	0
0		1	1	1	1		
1		0	1	1	0		

La función simplificada sería:  $S = B + \bar{C}$

A la hora de agrupar para sacar la función debemos hacer grupos de dos o cuatro y si no hay mas opcion tambien se puede hacer grupos de 1, siempre de en la misma fila o columna , nunca en diagonal, también se pueden formar cuadros eligiendo cifras de dos filas o columnas, como en la imagen.

## BIBLIOGRAFIA

<https://www.crcibernetica.com/n20-gearmotor-with-encoder-and-wheel/>

LM2596 DC DC

<https://www.youtube.com/watch?v=QPntXt8Ea3s>

Fuente DCDC LM2596

<https://www.youtube.com/watch?v=mXjmlwHC1IA>

Driver L293D

<https://sites.google.com/site/angmuz/proyecto-f2---control-motores-cc-con-l293d-y-la-icezum-alhambra>

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