



Esquemático y Simulación-Bicicleta Eléctrica

Alumnos: Elian Nevado, Gian Machicado, Alvaro Garcia, Heyder Rojas



-Esquemático y Simulación

- ESC
- Información de referencia
- Componentes pedidos
- Simulaciones
- Código para la simulación
- Extras
- Conclusiones
- Bibliografía



ESC

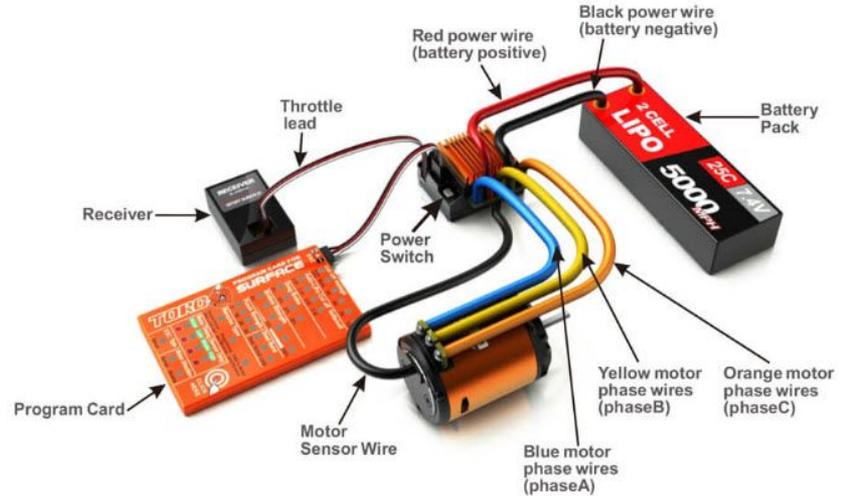
(Electronic Speed Controller)

Elian Nevado



Es un circuito electrónico que se utiliza para cambiar la velocidad de un motor eléctrico, su sentido y también puede funcionar como freno dinámico.

Estos se utilizan con frecuencia en modelos de control por radio que funcionan con energía eléctrica. Este se usa con mayor frecuencia para motores sin escobillas, en un sistema alimentado por una fuente de energía de bajo voltaje y trifásica producida electrónicamente para el funcionamiento del motor.



APLICACIONES DE ESC

Los sistemas electrónicos de control de velocidad se utilizan en aplicaciones de vehículos y control remoto.

- Coches eléctricos
- Aviones eléctricos
- Carros
- Helicópteros
- Aviones
- Barcos
- Cuadricópteros
- Bicicletas eléctricas



ESC CON ARDUINO UNO (AT MEGA 328)

En el presente proyecto, desarrollamos dos tipos de ESC, un circuito en donde prima el uso de drivers y otro en donde prima el uso de transistores.

En ambos circuitos diseñados se buscará obtener señales pwm provenientes de los terminales del motor, cuyo resultado esperado es un conjunto de ondas seno.

Para comprender el funcionamiento de la bicicleta eléctrica, se deben comprender los componentes principales, los cuales son un motor eléctrico, un banco de baterías , un circuito compuesto por drivers o un inversor trifásico compuesto por transistores. Además, cuenta con un controlador que consta de un arduino “uno” con un microprocesador el cual funciona en conjunto con un código arduino para el acondicionamiento de señales.

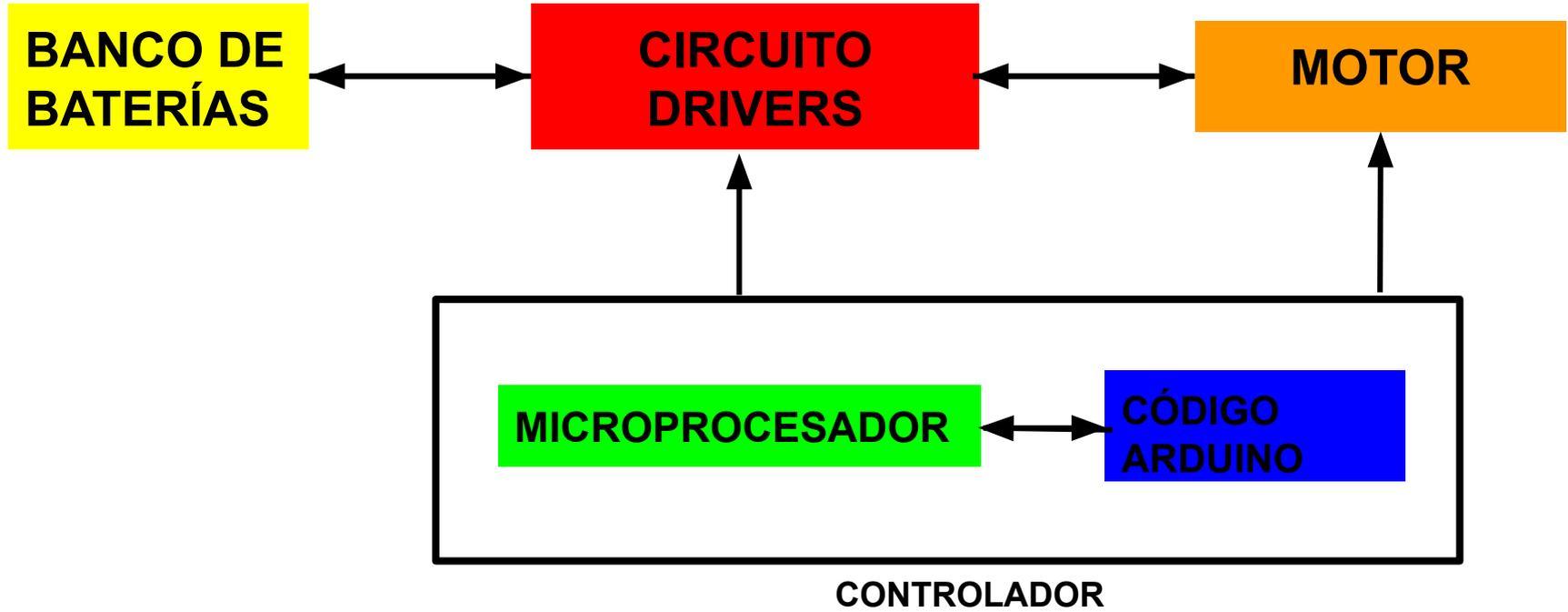
El funcionamiento de la bicicleta eléctrica básicamente consiste en la utilización de un banco de baterías, la cual alimenta al circuito y controlador. Este último aporta las señales que controlan la velocidad del motor a través de un potenciómetro. Los diagramas de bloques de la bicicleta son mostrados en los siguientes slides los cuales responden a lo antes planteado.

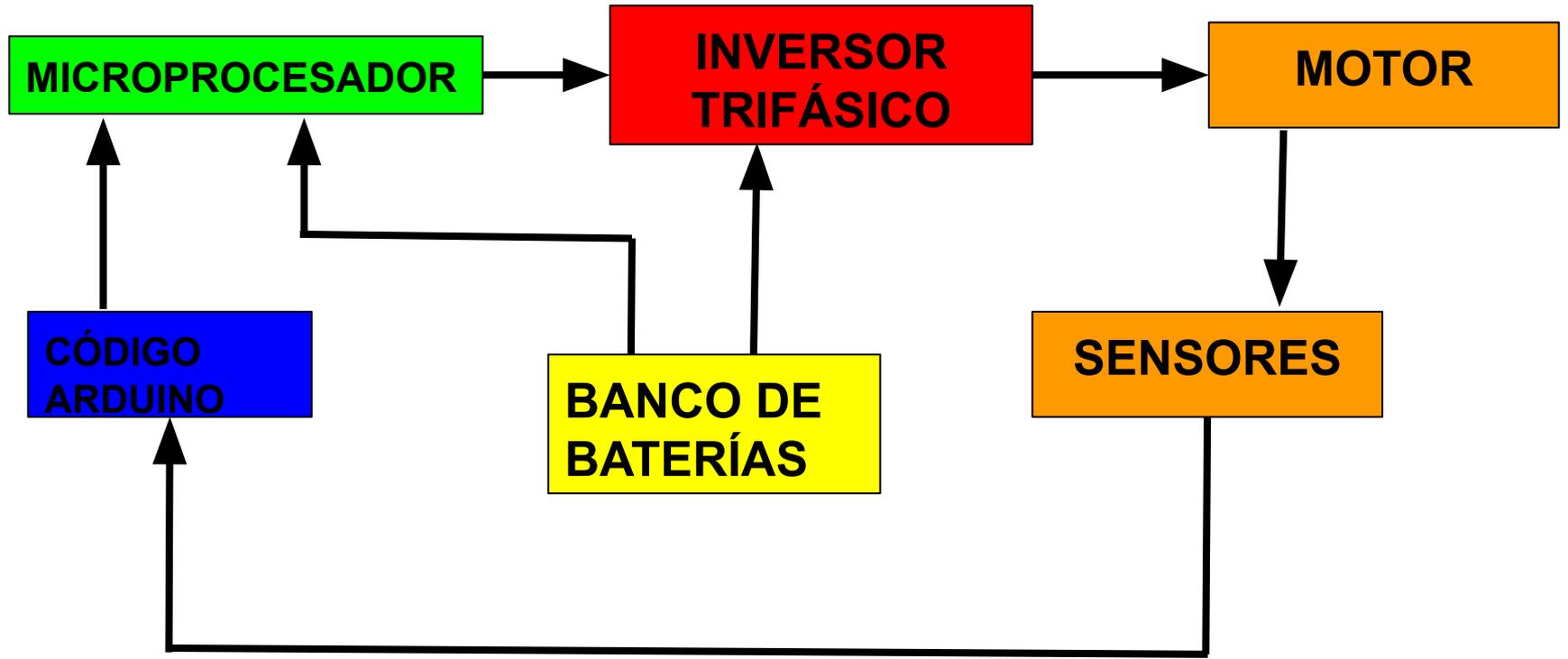


Diagrama de bloques

Elian Nevado







ESC con transistores BJT

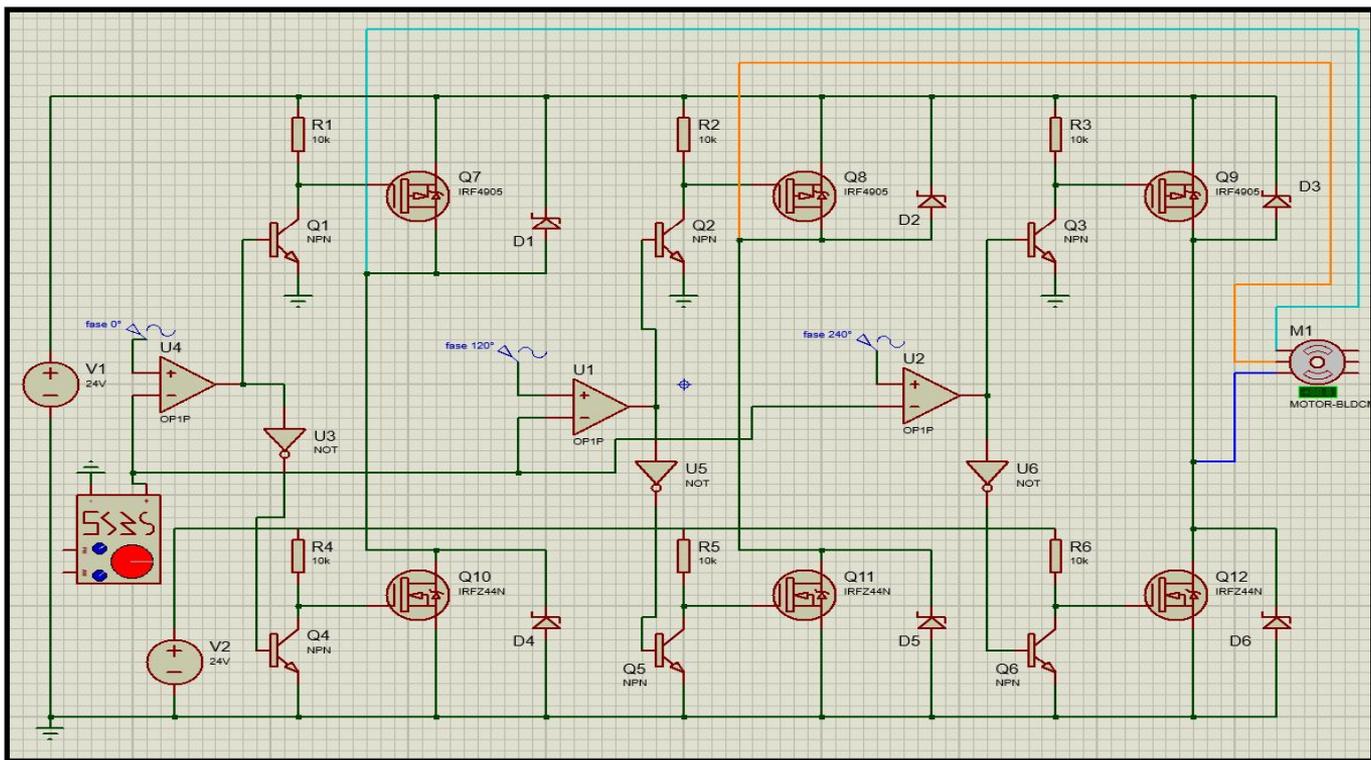
Heyder Rojas Jorge

componentes



- Motor Brushless DC (BLDC) de 24 V
- Resistor 6 x 10k ohm
- Fuente de alimentación de 24V
- MOSFET 3 x IRF4905
- MOSFET 3 x IRFZ44N
- Transistor BJT 6 x S8050
- Diodo Zener 6 x 1NS819

Esquemático



ESC con drivers

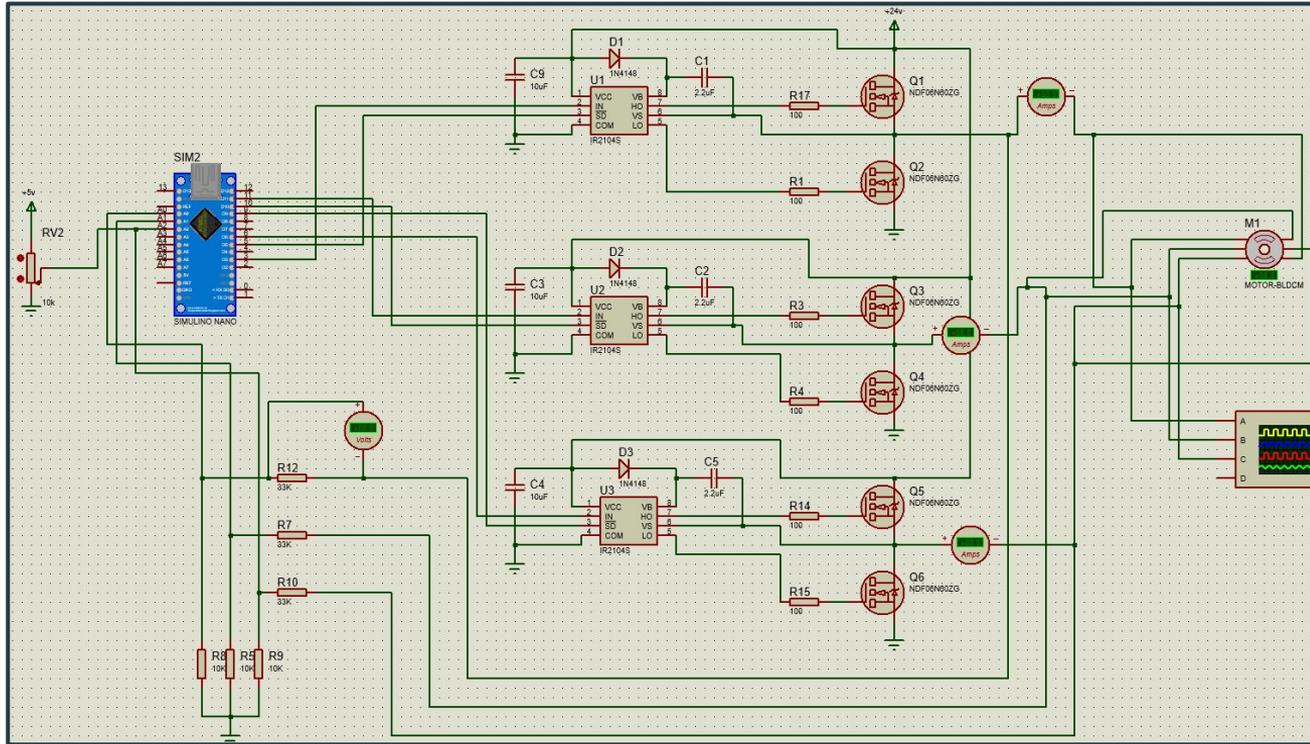
Alvaro Garcia

Componentes



- Arduino UNO
- Motor Brushless DC (BLDC) de 24 V
- Transistores tipo N 6 x 06N03LA
- Driver IC 3 x IR2104S (IR2104)
- Resistor 6 x 33k ohm
- Resistor 3 x 10k ohm
- Resistor 6 x 100 ohm
- Diodo 3 x IN4148
- Capacitor 3 x 10uF
- Capacitor 3 x 2.2uF
- Potenciómetro
- Fuente de alimentación de 24V
- Breadboard
- Jumper wires

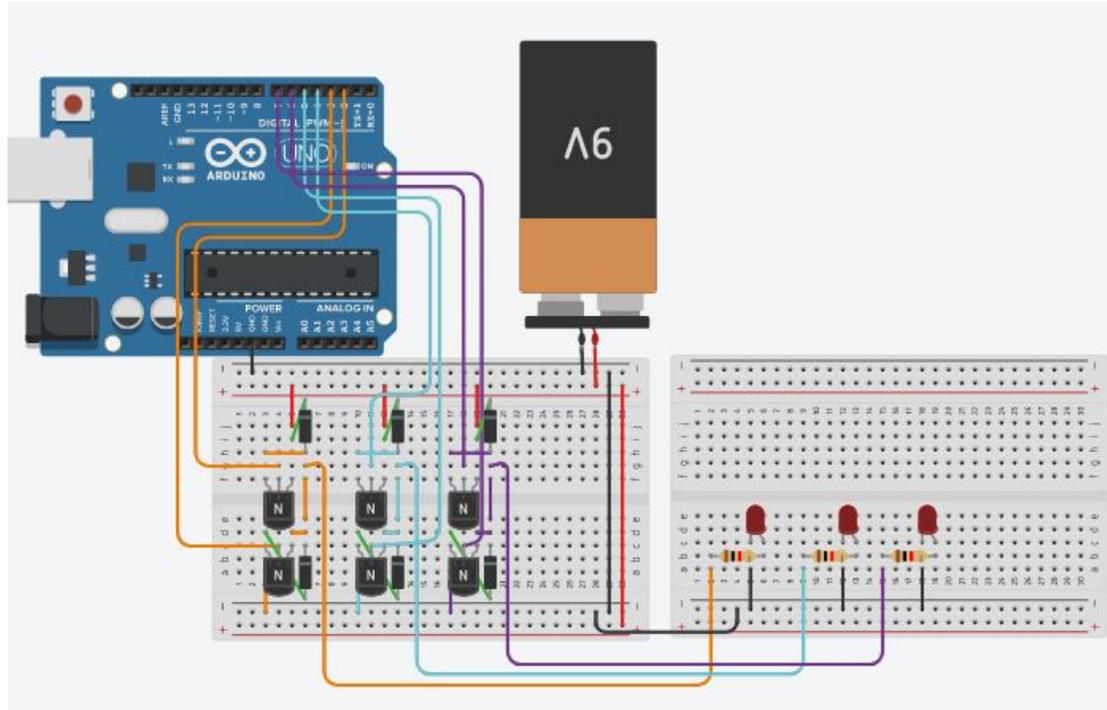
Esquemático



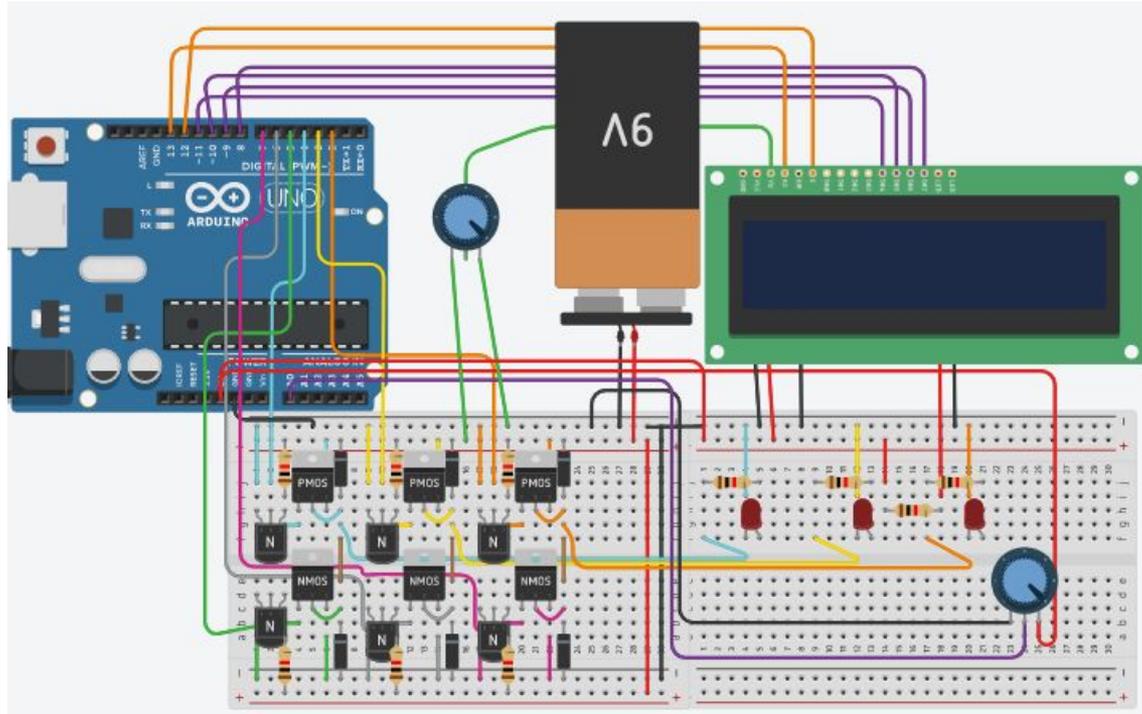
Simulaciones

Gian Daniel Machicado Condor
Heyder Rojas Jorge
Álvaro García Huatuco

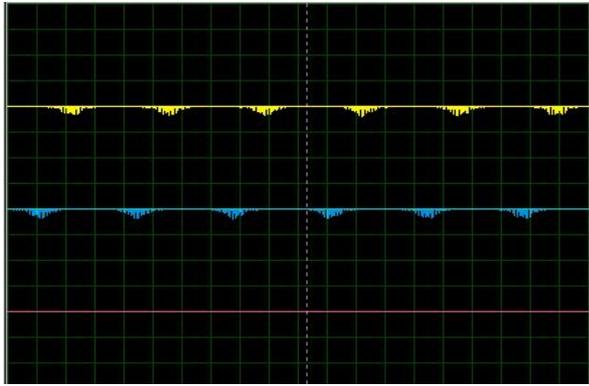
Simulación 1:



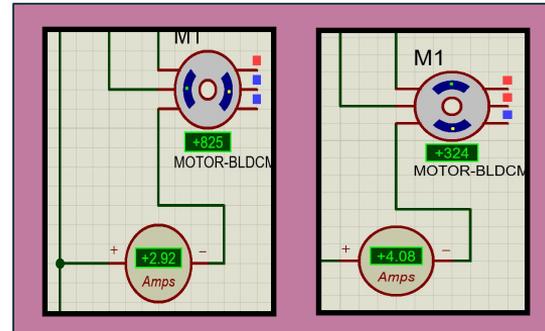
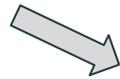
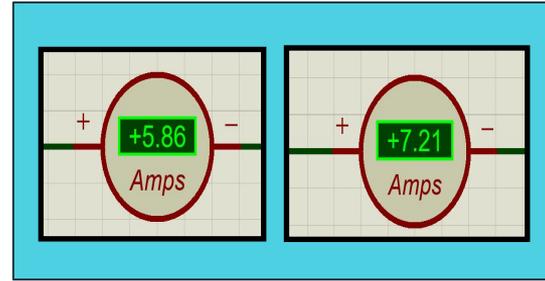
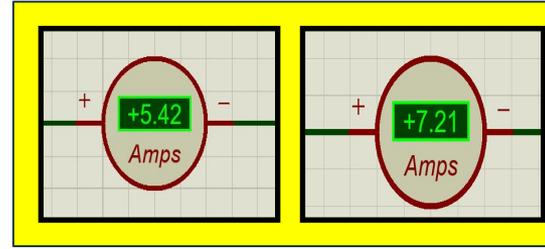
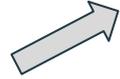
Simulación 2:

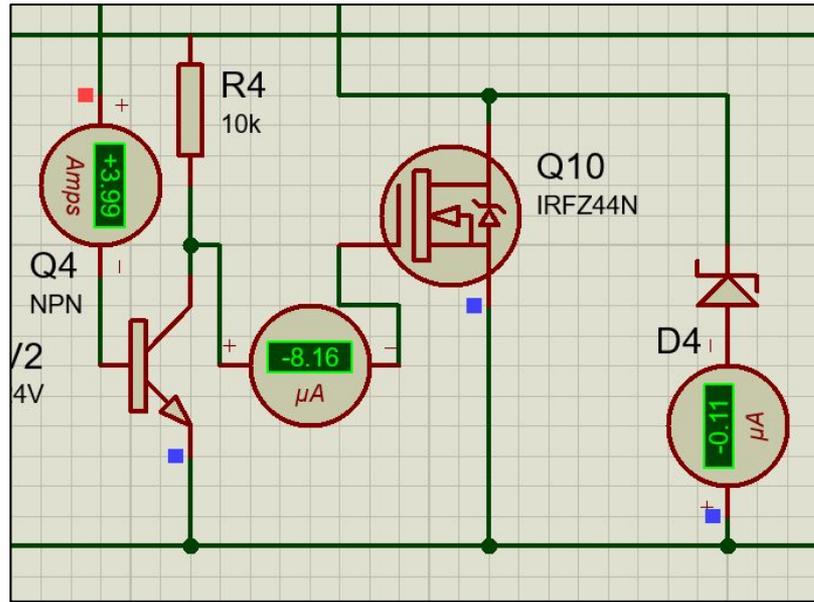


Simulación 3:



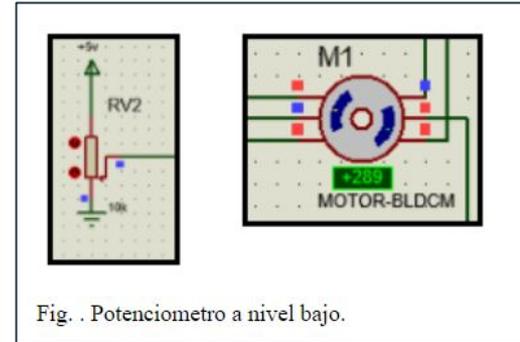
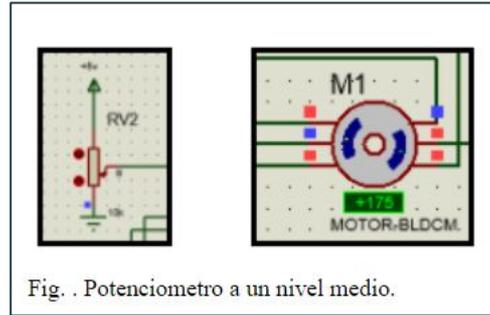
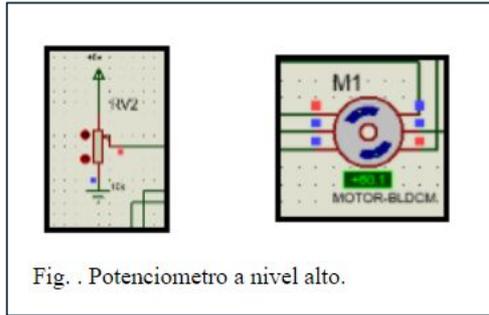
Ondas de fases del motor





Mediciones de corriente para algunos componentes

Simulación 4





Código para la simulación

Gian Daniel Machicado Condor



Asignacion de valores:

```
*/
```

Se controla la velocidad mediante un potenciómetro que esta en el pin A3

```
int AA1=3;  
int AA2=5;  
int BB1=11;  
int BB2=10;  
int CC1=9;  
int CC2=6;
```

El ESC siempre estara habilitado

```
int enable=2; //This is not used for now, The ESC is always enabled
```

```
int emfA=A0; Pines analogicos donde entrara la señal que es una fuerza  
int emfB=A1; electromotriz. Quiere decir que los pines analogicos reciban un voltaje  
int emfC=A2; como entrada.
```

```
int fase=1;  
int deltaA=0;  
int emA=0;  
int sum=0;
```

```
int IN=A3;  
int Delay=4000;
```

```
int it=0;  
int it2=1;
```

```
static int delta=0;  
static int Lastdelta=-1;
```

Delta y Last Delta se van a ir actualizando casi al final del código.

```
unsigned long previousMillis = 0;
```

Asignación de pines:

```
void setup() {  
  Serial.begin(250000);  
  pinMode(AA1,OUTPUT);  
  pinMode(AA2,OUTPUT);  
  pinMode(BB1,OUTPUT);  
  pinMode(BB2,OUTPUT);  
  pinMode(CC1,OUTPUT);  
  pinMode(CC2,OUTPUT);  
  
  pinMode(enable,OUTPUT);  
  
  pinMode(IN,INPUT);  
  pinMode(emfA,INPUT);  
  pinMode(emfB,INPUT);  
  pinMode(emfC,INPUT);  
  
  //digitalWrite(enable,HIGH);  
  //previousMillis = micros();  
  
}
```

Serial.begin() es una funcion que te permite modular la velocidad de entrega y recibo de informacion en baudios.

Un baudio esta conformado por bits donde se agrupan para formar simbolos. El Arduino

requiere recibir mucha info en poco tiempo por eso es que el numero es tan alto, lo normal e que se inialize con 9600.

Se setean las variables a los pines y se definen si daran o recibiran informacion.

Los A0, A1, A2 y A3 son pines digitales, el resto es digital.

Movimiento del motor:

```
void loop() {  
  
  int emA = analogRead(emfA);  
  int emB = analogRead(emfB);  
  int emC = analogRead(emfC);  
  int sum = (emA+emB+emC)/3;  
  
  unsigned long currentMillis = micros();  
  
  if(currentMillis - previousMillis >= Delay){  
  
    previousMillis += Delay;  
  
    //Phase1 C-B  
    switch(fase){  
    case 1:  
      digitalWrite(AA1,LOW);  
      digitalWrite(AA2,LOW);  
      digitalWrite(BB1,LOW);  
      digitalWrite(CC2,LOW);  
      digitalWrite(BB2,HIGH);  
      digitalWrite(CC1,HIGH);  
      delta = emA-sum;  
  
      break;
```

Mide el voltaje de cada pin analogico

Suma y divide los voltajes donde su suma deberia dar aproximadamente 0.

Guarda el tiempo desde que se inicio el Arduino en microsegundos, y llega hasta los 70 minutos luego de eso se hace 0 y repite.

Milisegundos de iniciado el Arduino-Milis anteriores >= 4000

Milis anteriores = Milis anteriores + 4000

Se inicializa Milis anteriores con 0 y se va actualizando

Esta parte le dice a las bobinas prenderse o pagarse segun el caso.

Ya que C y B se activaron, el sensor sera A. Recibe el voltaje de A lo resta con sum.

Donde sum es aproximadamente 0, solo quedaria A que va a dar el voltaje inducido por el iman a la bobina

El proceso es el mismo para los siguientes casos. Una vez termina el caso. Se usa el break para salir del switch.

Y seguir con las siguientes lineas de codigo. Despues del break salta hasta esta linea.

Movimiento del motor:

```
//Phase2 A-B
```

```
case 2:
```

```
digitalWrite(AA2,LOW);  
digitalWrite(BB1,LOW);  
digitalWrite(CC1,LOW);  
digitalWrite(CC2,LOW);  
digitalWrite(AA1,HIGH);  
digitalWrite(BB2,HIGH);  
delta = emC-sum;  
break;
```

```
//Phase3 A-C
```

```
case 3:
```

```
digitalWrite(AA2,LOW);  
digitalWrite(BB1,LOW);  
digitalWrite(BB2,LOW);  
digitalWrite(CC1,LOW);  
digitalWrite(CC2,HIGH);  
digitalWrite(AA1,HIGH);  
delta = emB-sum;  
break;
```

```
//Phase4 B-C
```

```
case 4:
```

```
digitalWrite(AA1,LOW);  
digitalWrite(AA2,LOW);  
digitalWrite(BB2,LOW);  
digitalWrite(CC1,LOW);  
digitalWrite(BB1,HIGH);  
digitalWrite(CC2,HIGH);  
delta = emA-sum;  
break;
```

```
//Phase5 B-A
```

```
case 5:
```

```
digitalWrite(AA1,LOW);  
digitalWrite(BB2,LOW);  
digitalWrite(CC1,LOW);  
digitalWrite(CC2,LOW);  
digitalWrite(AA2,HIGH);  
digitalWrite(BB1,HIGH);  
delta = emC-sum;  
break;
```

```
//Phase6 C-A
```

```
case 6:
```

```
digitalWrite(AA1,LOW);  
digitalWrite(BB1,LOW);  
digitalWrite(BB2,LOW);  
digitalWrite(CC2,LOW);  
digitalWrite(CC1,HIGH);  
digitalWrite(AA2,HIGH);  
delta = emB-sum;  
break;  
}
```

Movimiento del motor:

```
if (Lastdelta < 0){
if (delta > 0)
{
Lastdelta=delta; //save the last delta
fase=fase + 1;
if (fase > 6){
fase = 1;
}
}
} //Zero cross from - to +

if (Lastdelta > 0){
if (delta < 0)
{
Lastdelta=delta;
fase=fase + 1;
if (fase > 6){
fase = 1;
}
}
} //Zero cross from + to -

//Case ends
```

Se inicializa Last delta con -1 y delat con 0

Es exclusivamente para el prendido del motor, ya que luego last delta sera siempre mayor a 0.

Hace el cambio de fase despues de actualizar last delta. Cuando la fase sea mayor a 6 debera volver a 1 por que no existen casos 0 o 7.

Condicionales para movimiento:

```
int t = analogRead(IN); //From the potentiometer
Delay = map(t, 0, 1024, 1, 1000); //we obtain the delay speed using the potentiometer
//we map the values from 1 to 1000 microseconds
```

```
} //loop ends
```

Se setea la variable Delay con 4000 luego se actualiza con el potenciometro. Se setea la perilla del potenciometro desde 0 a 1024 Voltios y se mapea entre que 1 a 1000 que va a ser el nuevo delay. El delay es lo que hace que las bobinas se prendan o apagen. Determina la velocidad del apagado o prendido de las bobinas, es por eso que controla la velocidad del motor. donde 1 seria la maxima velocidad y 1000 seria la velocidad minima.

Batería

Alvaro Garcia Huatuco

Celda que se usará:

Baterías Li-ion 3,7V/3000mAh (voltaje del motor 24V)

Especificaciones de la celda:

Voltaje estándar: **3,6 V-3,7 V**

Voltaje de carga: 4,20 +/-0,03 V

Voltaje de corte de descarga: 3,0 V

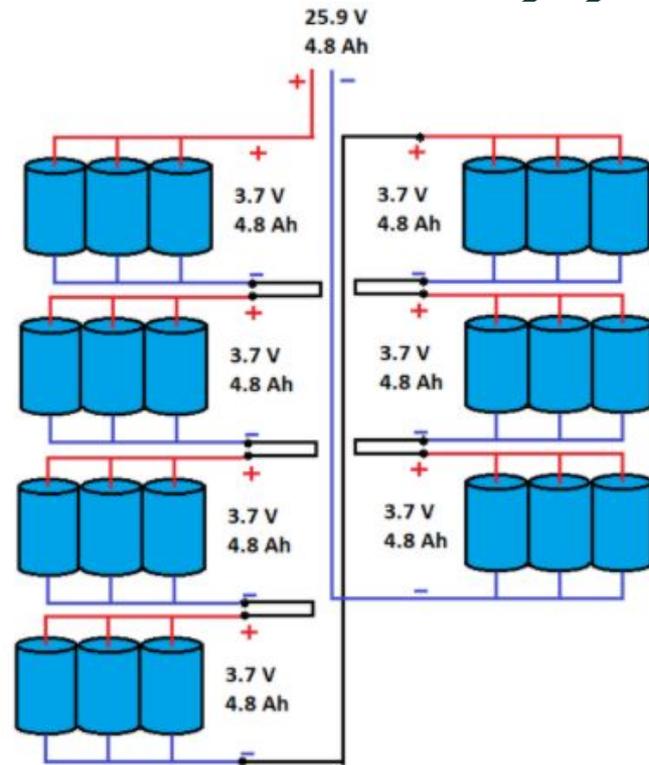
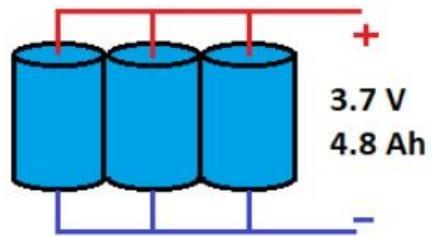
Peso de referencia: 45g-47g (1,66 oz) por celda

Longitud: 65mm, diámetro: 18mm

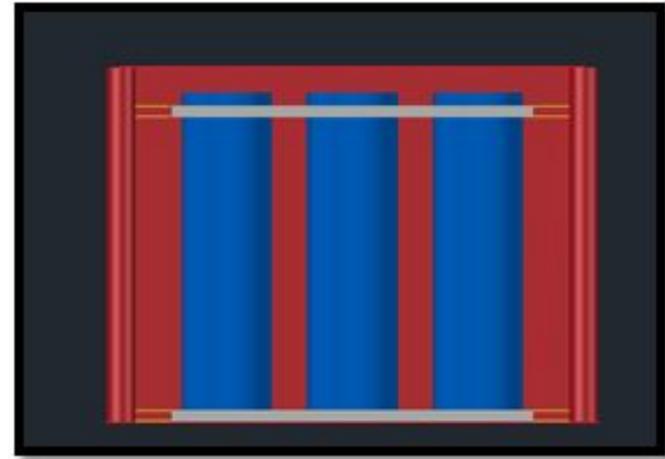
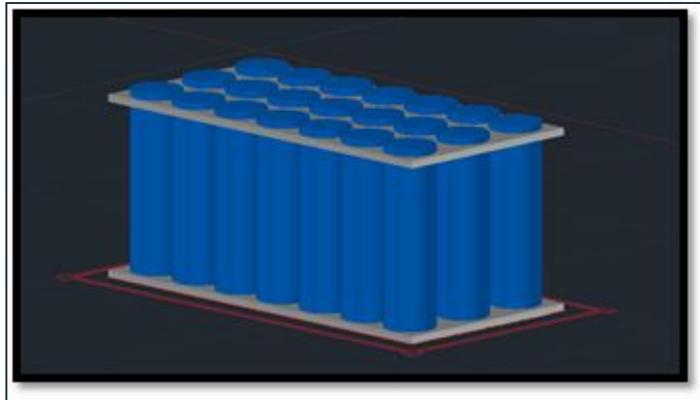
Para el desarrollo de nuestra batería:

7 EN SERIE Y 3 EN PARALELO obtendremos 24V 9AH

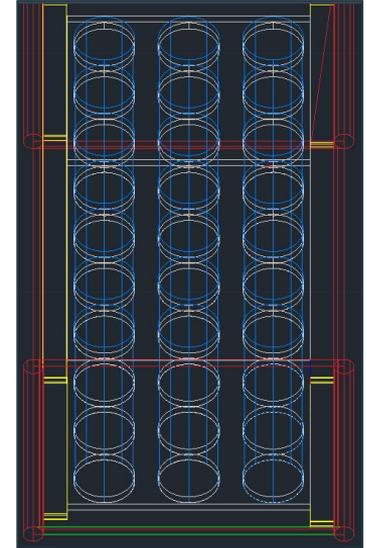
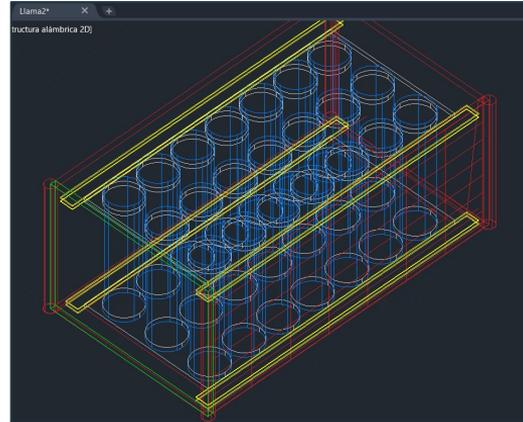
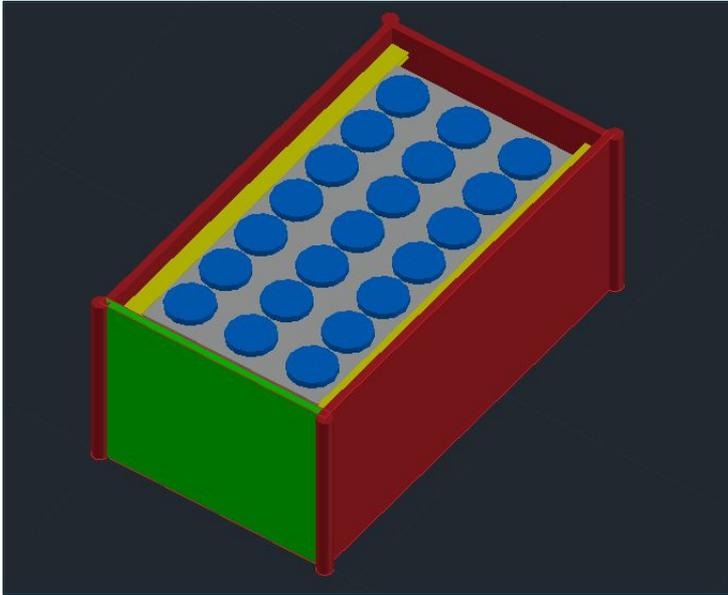




Diseño en autocad



Diseño en autocad



Dimensiones: 7 cm *9.2 cm*16.4cm

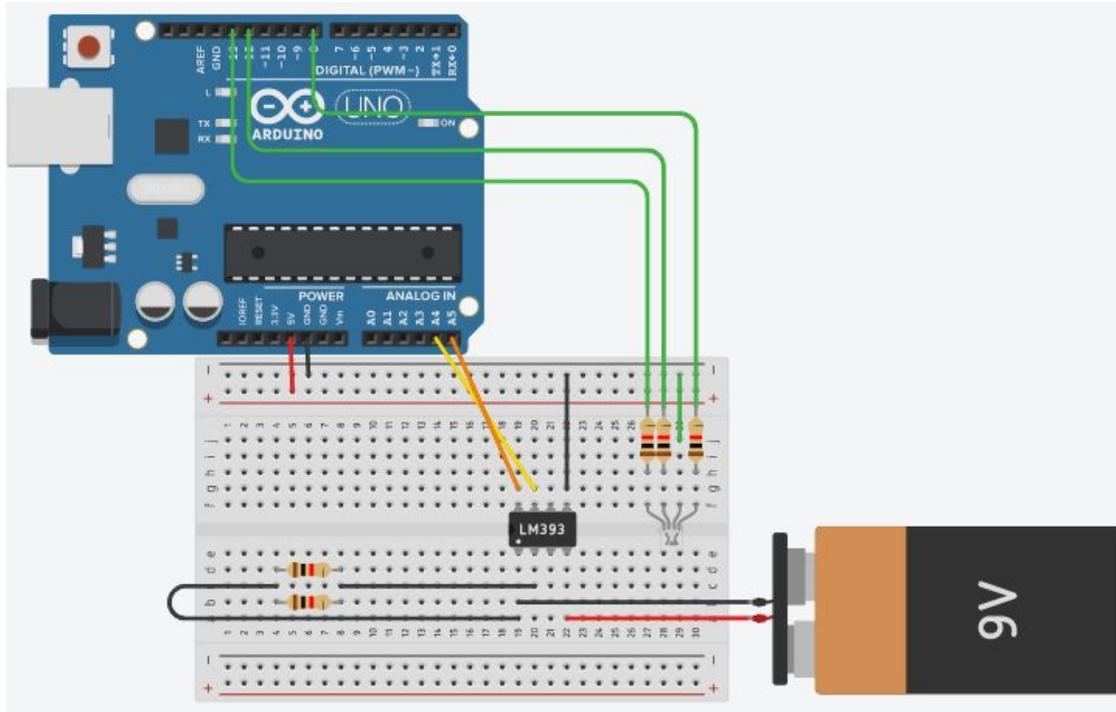


Extras

Gian Daniel Machicado Condor



Interfaz de usuario:



```
int verde=13;  
int azul=12;  
int rojo=8;  
float A;  
float V;
```

```
void setup()  
{  
  pinMode(rojo, OUTPUT);  
  pinMode(azul, OUTPUT);  
  pinMode(verde,  
OUTPUT);  
  Serial.begin(9600);  
  pinMode(A0, INPUT);  
  pinMode(A1, INPUT);  
}
```

```
void loop()  
{  
  V= analogRead(A5)/41.2;  
  A= analogRead  
(A4)/183.3;  
  Serial.print(V);  
  Serial.print(" V");  
  Serial.print("\t");  
  Serial.print(A);  
  Serial.println(" A");
```

```
if (24>V && V>22){  
  digitalWrite(rojo, LOW);  
  digitalWrite(azul, LOW);  
  digitalWrite(verde, HIGH);  
  delay(10000);  
  //150 mA de consumo  
  digitalWrite(rojo, LOW);  
  digitalWrite(azul, LOW);  
  digitalWrite(verde, LOW);  
  delay(100 000);  
}
```

```
else if(V>20.8){  
  digitalWrite(rojo, HIGH);  
  digitalWrite(azul, LOW);  
  digitalWrite(verde, HIGH);  
  delay(10000);  
  digitalWrite(rojo, LOW);  
  digitalWrite(azul, LOW);  
  digitalWrite(verde, LOW);  
  delay(100 000);  
}
```

```
else{  
  digitalWrite(rojo, HIGH);  
  digitalWrite(azul, LOW);  
  digitalWrite(verde, LOW);  
  delay(10000);  
  digitalWrite(rojo, LOW);  
  digitalWrite(azul, LOW);  
  digitalWrite(verde, LOW);  
  delay(100 000);  
}
```

Conclusiones

Gian Daniel Machicado Condor
Heyder Rojas Jorge

ESC con transistores BJT:

1. Se observa un correcto funcionamiento en la simulación obviando los instrumentos de medición para el voltaje y corriente de algunos componentes.
2. La corriente que pasa por el motor Brushless oscila entre aproximadamente 2 a 4 A.
3. Las señales del motor no son las esperadas.
4. Antes de probar el circuito con el arduino se utilizaron OPAMPs como primeras pruebas.

ESC con drivers:

1. El motor funciona de acuerdo a el código, hace los respectivos cambios junto a las señales del potenciómetro
2. El código proporcionado es simple, conciso y funcional. Entenderlo no toma mucho tiempo, y es escalable.
3. El código usa 8 pines Digitales y 4 pines Analógicos; dejando 2 pines de Bluetooth, 4 pines Digitales y 2 pines Analógicos. Por lo que se pueden añadir futuras mejoras a la bicicleta.



Bibliografía

BIBLIOGRAFÍA:

Anónimo. (14 de Abril de 2016). *Bikelec*. Obtenido de Bikelec: <https://www.bikelec.es/blog/motor-central-vs-motor-rueda/>

Anónimo. (s.f.). *bhbikes*. Obtenido de bhbikes:

https://www.bhbikes.com/es_ES/bicicletas-electricas/bicicletas-electricas-gravel/core

Anónimo. (s.f.). *bhbikes*. Obtenido de bhbikes:

https://www.bhbikes.com/es_ES/bicicletas-electricas/bicicletas-electricas-gravel/core-carbon

Anónimo. (s.f.). *bhbikes*. Obtenido de bhbikes:

https://www.bhbikes.com/es_ES/bicicletas-electricas/bicicletas-electricas-gravel/core-gravelx-2-2

Anónimo. (s.f.). *Lovesharing*. Obtenido de Lovesharing: <https://www.lovesharing.com/tipos-bicicletas-electricas/>

González, A. (17 de Agosto de 2020). *HÍBRIDOS Y ELÉCTRICOS*. Obtenido de HÍBRIDOS Y ELÉCTRICOS:

<https://www.hibridosyelectricos.com/articulo/bicicletas-electrica/tipos-bicicletas-electricas-cual-es-mejor/20200811193038037317.html>

Roll, B. &. (22 de 10 de 2018). *tallerbicicletasmadrid*. Obtenido de tallerbicicletasmadrid:

<https://www.tallerbicicletasmadrid.com/blog/motor-kit-bicicleta-rueda-delantera-trasera-central-cual-es-mejor>

Anónimo. (s.f.). *amazon.es*. Obtenido de amazon.es:

https://www.amazon.es/Xian-conversi%C3%B3n-Trasera-Bicicleta-el%C3%A9ctrica/dp/B081RPSWPQ/ref=sr_1_4?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=motor+de+bicicleta+electrica+250w&qid=1607480231&sr=8-4

Anónimo. (s.f.). *tienda.bricogeek*. Obtenido de tienda.bricogeek:

<https://tienda.bricogeek.com/descatalogado/325-motor-brushless-1000kv-outrunner.html#:~:text=Resistencia%20interna%3A%2090%20m%CE%A9,Peso%3A%2047%20gramos>

Anónimo. (s.f.). *wellride.cl*. Obtenido de wellride.cl:

<https://wellride.cl/tienda-wellride/pinon-de-7-velocidades-11-24-black-compatible-con-shimano/>

1. Andina.(2020). Solo el 3 % de limeños usa la bicicleta de manera regular, revela estudio. Recuperado de: [https://andina.pe/agencia/noticia-solo-3-limenos-usa-bicicleta-manera-regular-revela-estudio-802250.aspx#:~:text=06%3A57%20%7C%20Lima%2C%20jun,y%20Opini%C3%B3n%20P%C3%BAblica%20\(CPI\).](https://andina.pe/agencia/noticia-solo-3-limenos-usa-bicicleta-manera-regular-revela-estudio-802250.aspx#:~:text=06%3A57%20%7C%20Lima%2C%20jun,y%20Opini%C3%B3n%20P%C3%BAblica%20(CPI).)
2. Andina. (2020).Lima Cómo Vamos: 20 % de viajes urbanos podría hacerse en bicicleta tras cuarentena. Recuperado de: <https://andina.pe/agencia/noticia-lima-como-vamos-20-viajes-urbanos-podria-hacerse-bicicleta-tras-cuarentena-801032.aspx>
3. ELECTRONOBS España. (2020). AprendeMás#7 - Los ESC | Motores Brushless | Toda La Teoría Que Necesitas Saber. Recuperado de: <https://www.youtube.com/watch?v=5HxUQvt3R4E>
4. ELECTRONOBS España. (2018). Pr#002 - Construir un ESC para motores sin escobillas sin sensores. Recuperado de: <https://www.youtube.com/watch?v=FZKbt8HIT8Q>
5. ELECTRONOBS España. (2019). Pr#76 - ESC Casero Con Arduino - Código Abierto. Recuperado de: <https://www.youtube.com/watch?v=0ID2rNpDicU>
6. ELECTRONOBS. (2018). DIY Sensored ESC - full tutorial. Recuperado de: <https://www.youtube.com/watch?v=G9PHqN9vmVI&list=LL&index=4>
7. ELECTRONOBS. (2018). DIY Sensored ESC - full tutorial. Recuperado de: http://www.electrionoobs.com/eng_circuitos_tut19_sch2.php
8. ELECTRONOBS. (2018). DIY Sensored ESC - full tutorial. Recuperado de http://www.electrionoobs.com/eng_circuitos_tut19_sch1.php