



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y DISEÑO INDUSTRIAL
INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA
REGULACIÓN AUTOMÁTICA



Proyecto control de motor de corriente continua

Eduardo Arcones del Álamo

Nº matrícula: 51.085



Índice

Introducción	3
Material	4
Descripción de elementos	5
Motor de Corriente continua	5
Circuito integrado L293D (Doble puente en h)	6
Fotodiodo 2030	10
Análisis teórico	11
Caso 1: $PIN_1 = PIN_2 = APAGADO$	12
Caso 2 $PIN_1 = PIN_2 = ACTIVO$	13
Caso 3 $PIN_1 = ACTIVO$ $PIN_2 = APAGADO$	14
Caso 4: $PIN_1 = APAGADO$ $PIN_2 = ACTIVO$	15
Modelado del circuito	16
Matlab	23
Montaje	24
Programa	27
Observaciones	35



Introducción

El fin de este proyecto es la automatización del control de velocidad y de giro de un motor de corriente continua, de tal manera que, ante la entrada de la velocidad y el sentido de giro deseado, el circuito sea capaz de autorregularse para conseguirlo.

Con esto conseguiremos desde la capacidad de control del movimiento de persianas, hasta el control del motor de un coche.

La placa con la que realizaremos el circuito será una *Arduino UNO*, acompañada de un *punte en h* para poder variar el sentido de giro del motor.

El motor a mover es un *motor de corriente continua de 6v*, el cual sobrealimentamos a 9v para poder ver mejor la variación de velocidad.

Para saber la velocidad del motor construiremos un tacómetro a partir de un *fotodiodo* y un *diodo emisor de Luz (LED)*.

El tacómetro, controlado por la Arduino, obtendrá la velocidad del motor, que comparándola con la información introducida en el serial, hará que se aumente o disminuya la tensión del motor, variando con esto la velocidad. Sabiendo que pin está activado y cual desactivado, nos informará también de si el motor está girando en sentido horario o antihorario.



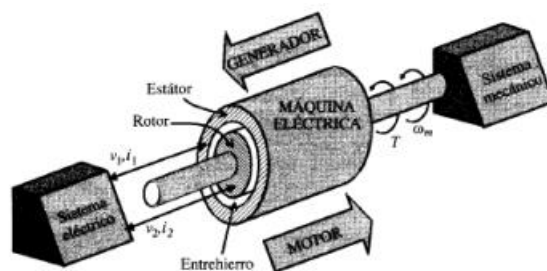
Material

- Arduino Uno
- Circuito integrado L293D (doble puente en h)
- Fotodiodo 2030
- Diodo Led
- Fuente externa de 9v
- Protoboard
- Jumpers
- Resistencias de 220Ω

Descripción de elementos

Motor de Corriente continua

Un motor de corriente continua es una máquina eléctrica rotativa. Como tal puede funcionar en indistintamente como generador o como motor(es reversible):



Todas las máquinas eléctricas rotativas constan de dos partes:

Estator → Parte fija de la máquina

Rotor → Parte giratoria.

Como tiene tanto medio mecánico como eléctrico, el entrehierro supone el medio de acoplamiento entre estos.

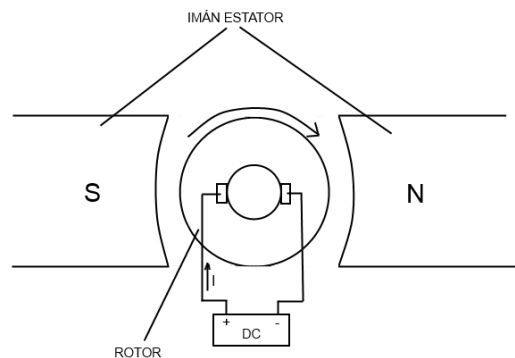
Tanto el rotor como el estator poseen devanados para poder crear el flujo que circula por el entrehierro. Podemos distinguir entre:

- Devanado inductor → es el que crea un flujo en el entrehierro
- Devanado inducido → en él se inducen corrientes para la variación del flujo

Pueden estar indistintamente en el estator o en el rotor, en función de la facilidad constructiva, aislamiento...

Cuando la corriente eléctrica circula por el devanado del rotor, se crea un campo electromagnético. Este interactúa con el campo magnético del imán del estator. Esto deriva en un rechazo entre los polos del imán del estator y del rotor creando un par de fuerza donde el rotor gira en un sentido de forma permanente.

Si queremos cambiar el sentido de giro del rotor, tenemos que cambiar el sentido de la corriente que le proporcionamos al rotor; basta con invertir la polaridad de la pila o batería.

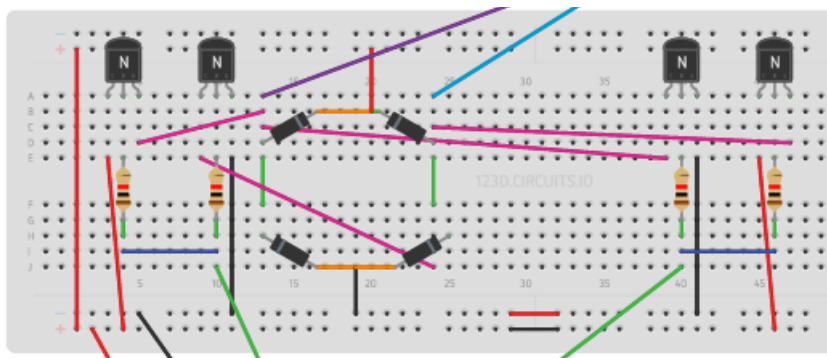


Circuito integrado L293D (Doble puente en h)

Para controlar un motor DC desde Arduino, tendremos que usar un driver para motores para proporcionarle más corriente al motor ya que las salidas del Arduino sólo dan 40mA. De esta manera, con el driver podemos alimentar el motor con una fuente de alimentación externa.

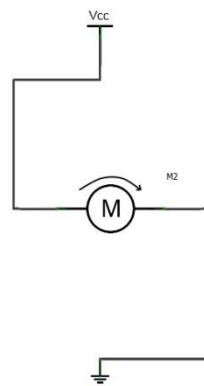
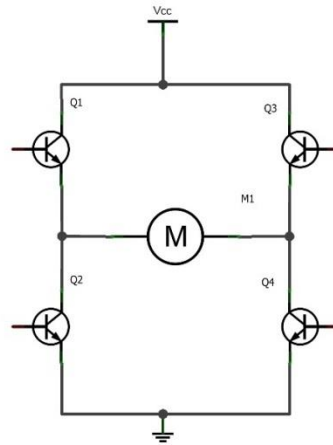
El L293D es un integrado para controlar motores DC que usa el sistema puente en H.

Un *puente en h* es un sistema para controlar el sentido de giro de un motor DC usando cuatro transistores. Los transistores se comportan como interruptores (relés) siendo activados por la placa dos a dos, para que la corriente pueda circular en un sentido o en otro sobre el motor

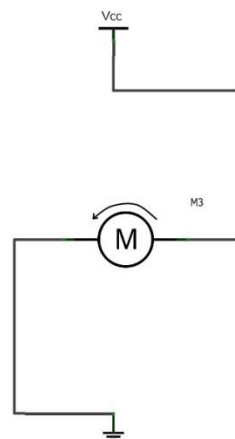




UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y DISEÑO INDUSTRIAL
INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA
REGULACIÓN AUTOMÁTICA

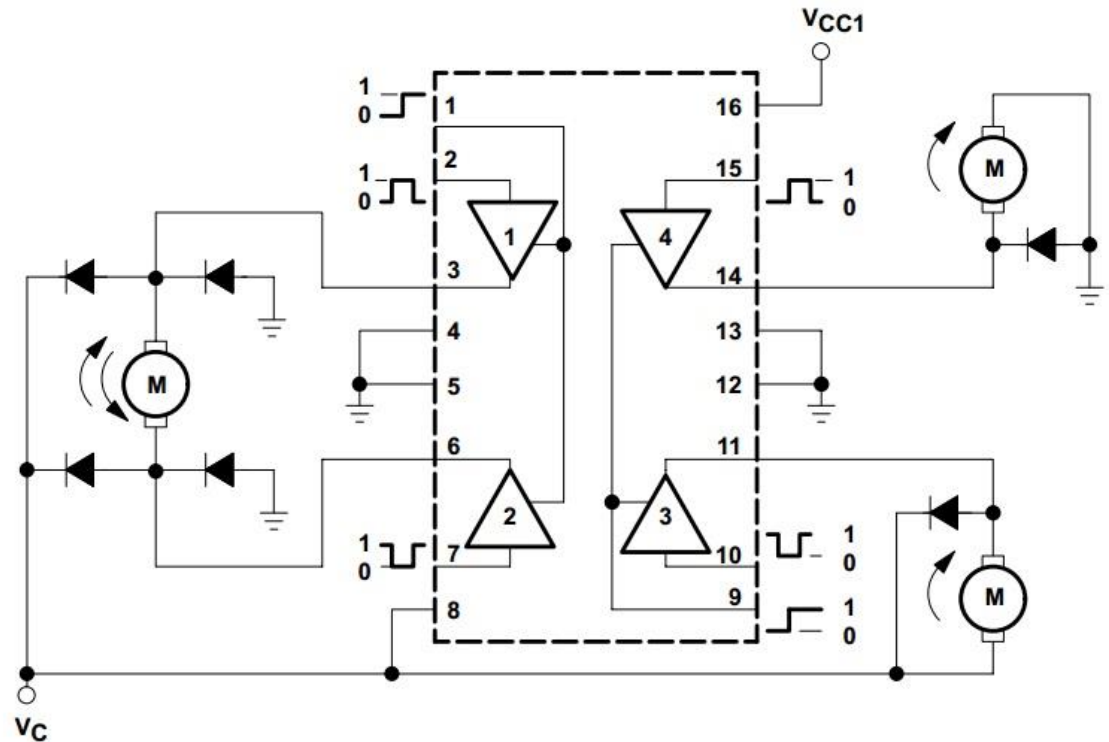


Q1 y Q4 conducen
Q2 y Q3 en corte



Q2 y Q3 conducen
Q1 y Q4 en corte

El L293D tiene dos puentes H y proporciona 600mA al motor y soporta un voltaje entre 4,5V y 36V



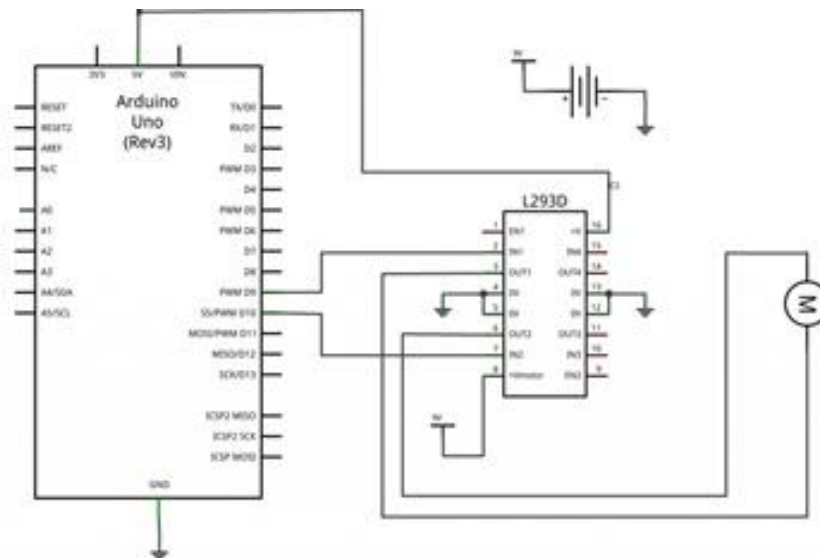
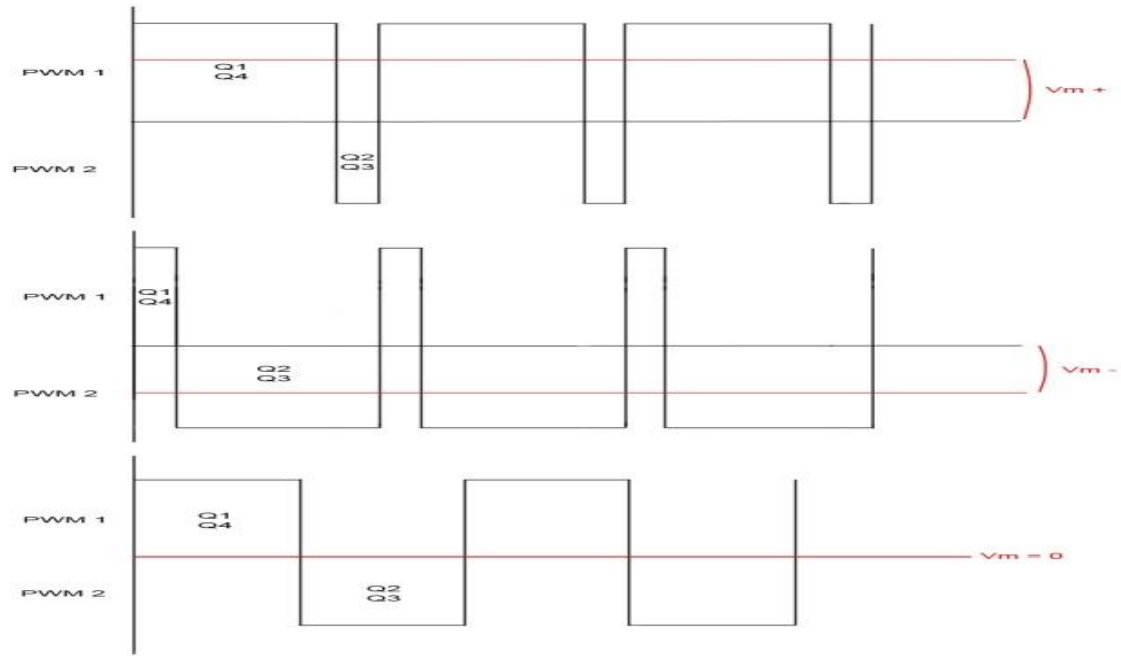
Emplearemos el puente en h de la parte izquierda. Los diodos que se encuentran externos al L293D, en realidad están en su interior.

Los pines 3 y 6 se conectan al motor, mientras que los pines 2 y 7 serán las entradas donde conectar las salidas del Arduino. Estos dos últimos harán que el motor gire en un sentido o en el otro, dependiendo de cuál de los dos se encuentre activado (ya que todo el circuito está montado para que se active con niveles altos).

Los diodos que rodean al motor se colocan a modo de protección, para evitar que al dejarle de llegar tensión, corrientes parasitas salgan de él hacia el integrado y la placa y los quemen.



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y DISEÑO INDUSTRIAL
INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA
REGULACIÓN AUTOMÁTICA



Fotodiodo 2030

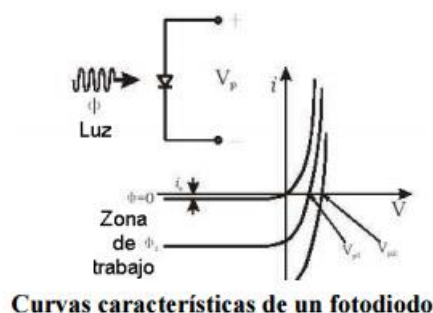
El fotodiodo se parece mucho a un diodo semiconductor común, con la diferencia de que es un dispositivo que conduce una cantidad de corriente eléctrica proporcional a la cantidad de luz que incide sobre él.



Esta corriente eléctrica fluye en sentido opuesto a la flecha del diodo y se llama corriente de fuga.

El fotodiodo se puede utilizar como dispositivo detector de luz, pues convierte la luz en electricidad y esta variación de electricidad es la que se utiliza para informar que hubo un cambio en el nivel de iluminación sobre el fotodiodo.

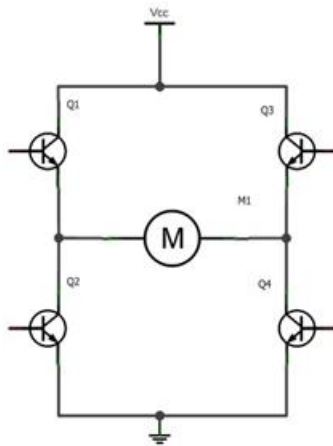
Si el fotodiodo es polarizado en directa, la luz que incide no tendría efecto sobre él y se comportaría como un diodo semiconductor normal. (Recuerde, el fotodiodo trabaja en inversa). La mayoría de los fotodiodos vienen equipados con un lente que concentra la cantidad de luz que lo incide, de manera que su reacción a la luz sea más evidente. A diferencia del LDR o fotorresistencia, el fotodiodo responde a los cambios de oscuridad a iluminación y viceversa con mucha más velocidad, y puede utilizarse en circuitos con tiempo de respuesta más pequeño.



Curvas características de un fotodiodo

Análisis teórico

Como hemos dicho el puente H se usa para invertir el giro de un motor, pero también puede usarse para frenarlo (de manera brusca), al hacer un corto entre los bornes del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia cuando desconectamos el motor de la fuente que lo alimenta. En el siguiente cuadro se resumen las diferentes acciones.



Q ₁	Q ₂	Q ₃	Q ₄	Acción
0	0	0	0	El motor se detiene bajo su inercia
1	1	1	1	El motor frena(fast-stop)
1	0	0	1	El motor gira en sentido horario
0	1	1	0	El motor gira en sentido antihorario

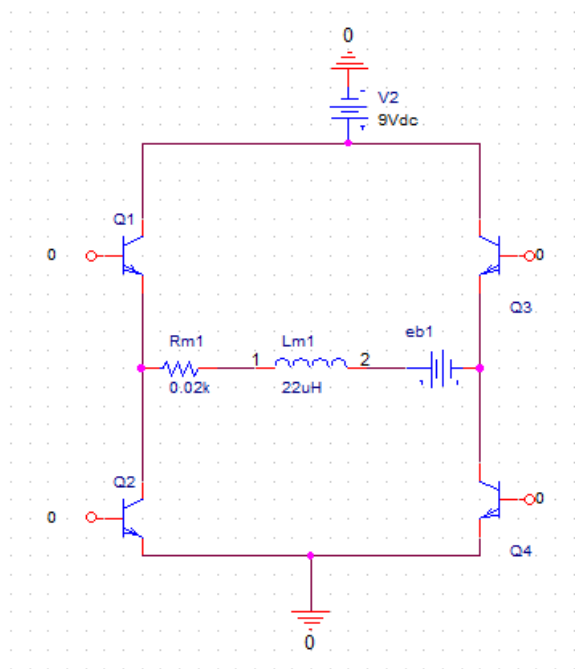
Procedemos a analizar los distintos casos:

Caso 1: $PIN_1 = PIN_2 = APAGADO$

En este caso las bases se encontrarían a 0v, por lo que los transistores estarían en zona de corte (ya que las tensiones de los emisores nunca podrán ser negativas).

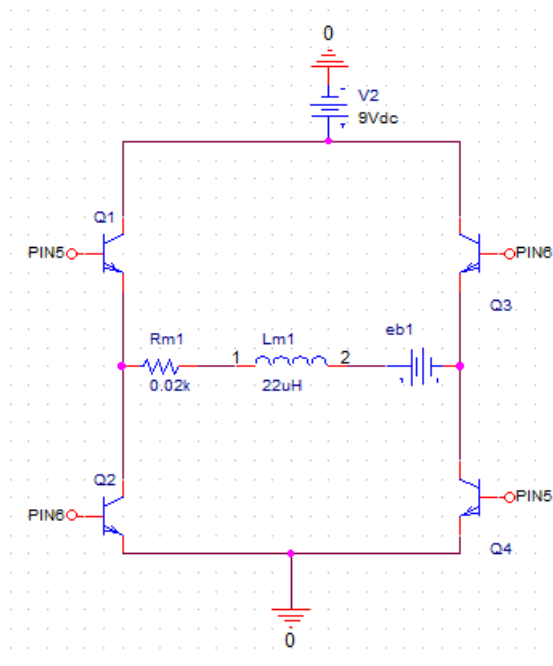
Esto implicaría que el motor pasaría a funcionar como generador, ya que entre sus polos habría la diferencia de tensión que el generase, y en el circuito la corriente que el generara.

Se detendría por su propia inercia.



Caso 2 $PIN_1 = PIN_2 = \text{ACTIVO}$

En este caso el motor estaría en cortocircuito, ya que ambos pines tendrían la misma tensión, por lo que su diferencia de tensión sería 0, lo que provocaría que se bloquease el rotor, originando una parada brusca.



Sabemos que el Arduino proporciona 40mA de corriente en el puerto, y que los transistores están actuando con ganancia de 15, por lo que las corrientes serán:

$$I_{B_1}(t) = I_{B_2}(t) = I_{B_3}(t) = I_{B_4}(t) = 40mA$$

$$I_{C_1}(t) = I_{C_3}(t) \approx 600mA$$

$$I_{E_1}(t) \approx I_{C_2}(t) \approx 600mA$$

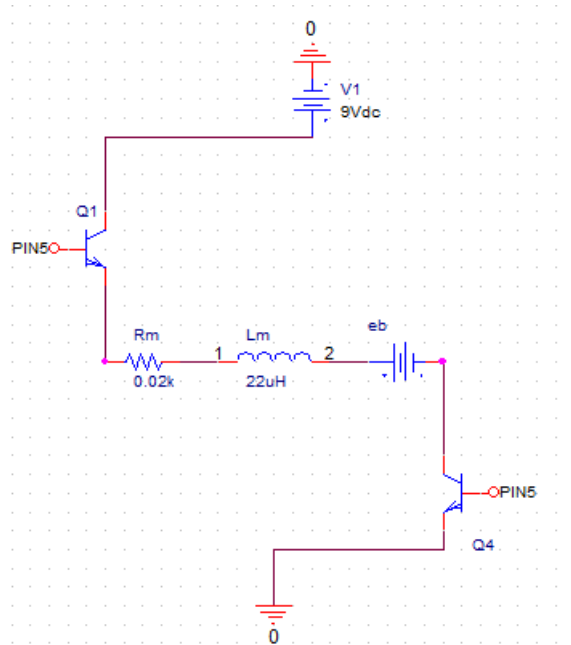
$$I_{E_3}(t) \approx I_{C_4}(t) \approx 600mA$$

$$I_{E_2}(t) = I_{E_4}(t) \approx 600mA$$

Aunque las corrientes de colector y emisor no son del todo iguales, podremos suponerlas iguales, ya que el error cometido por esto será del 3%.

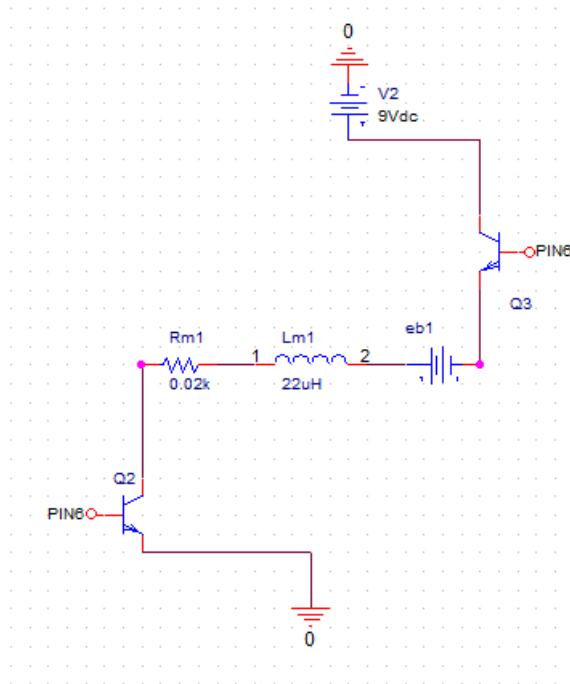


Caso 3 PIN₁=ACTIVO PIN₂=APAGADO



En este caso el motor giraría en sentido horario, al estar en saturación los transistores 1 y 4.

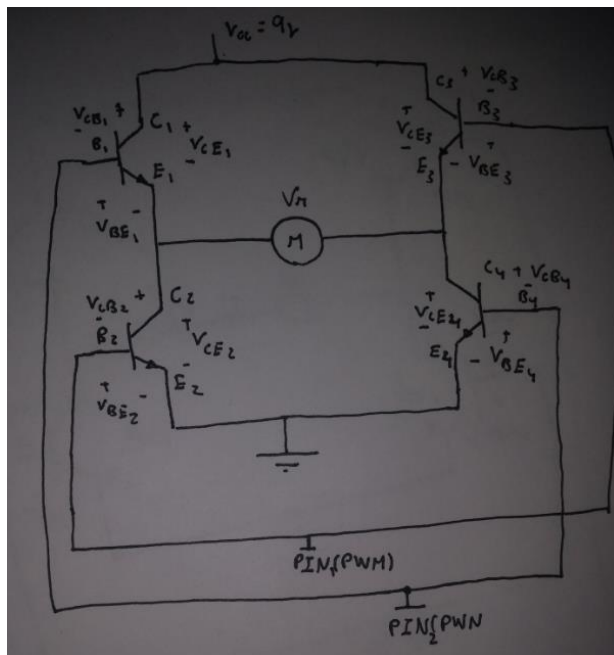
Caso 4: PIN₁=APAGADO PIN₂=ACTIVO



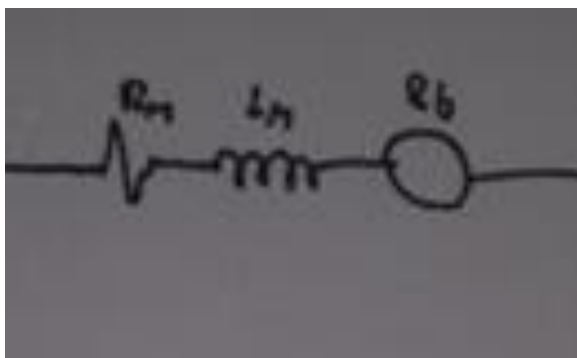
En este caso el motor giraría en sentido horario, al estar en saturación los transistores 2 y 3. La variación del sentido de giro se debe a que ahora la intensidad recorre el motor en sentido contrario al anterior, por la que el campo creado por la bobina irá en sentido opuesto al otro, haciendo que el sentido del motor sea el opuesto.

Modelado del circuito

Para saber la tensión que cae en el motor en los casos 3 y 4, como tenemos que:



Realizando el equivalente del motor:



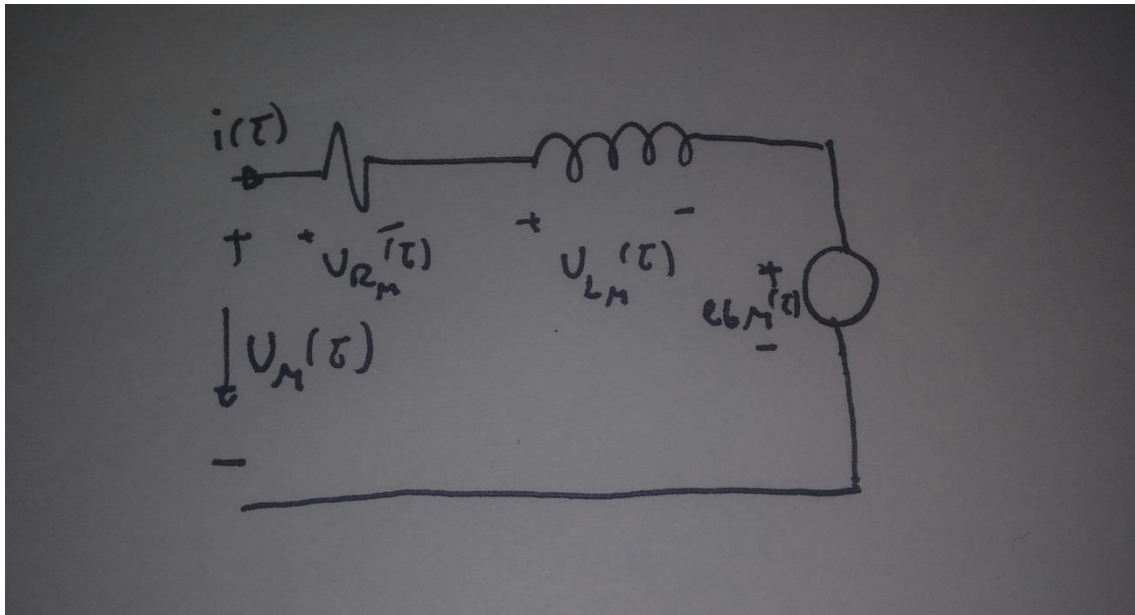
Sabemos que la resistencia es de 20 ohmios, la bobina es de 22mH y la constante de velocidad es de 827rpm/V:

$$R_M = 20\Omega$$

$$L_M = 22mH = 22 \cdot 10^{-3}H$$

$$k_b = \frac{1}{\text{constante de velocidad}} = \frac{1}{827} = 1.2 \cdot 10^{-3} \frac{V}{rpm}$$

Conociendo esto podemos pasar a obtener las ecuaciones del motor:



$$U_M(t) = U_{RM}(t) + U_{LM}(t) + e_{bm}(t)$$

Sabemos que la tensión que cae en todo el motor es de 8.6V (ya que la tensión V_{CE} de cada transistor en saturación será de 0.2V), y que se le suministra una intensidad de 600mA

$$U_{LM}(t) = L \cdot \frac{d[i(t)]}{dt}$$

$$U_{RM}(t) = i(t) \cdot R_M$$

$$e_{bm}(t) = k_b \cdot w_M(t)$$

Para poder regular las revoluciones del motor contamos con un tacómetro óptico realizando mediante un fotodiodo y un diodo Led. Este transforma los pulsos de luz que le llegan gracias a las revoluciones del motor (momentos de luz-no luz) en tensión de señal cuadrada, con la siguiente fórmula:

$$U_T(t) = k_T \cdot w_M(t)$$



Sabemos que la constante tacométrica k_T es $0,01 \frac{V}{\frac{rev}{s}}$

Pasándola a V/rpm:

$$k_t = 0,01 \frac{V}{\frac{rev}{s}} = 1.05 \cdot 10^{-3} \frac{V}{rpm}$$

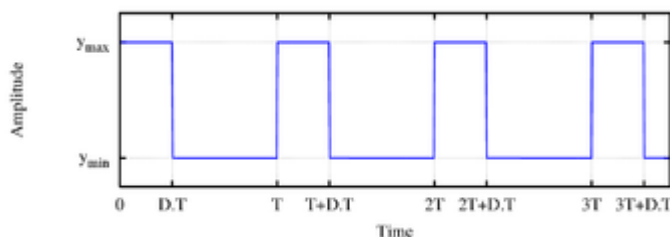
El sistema comparará las revoluciones medidas con las revoluciones deseadas, y aumentará o disminuirá la tensión entregada al motor para igualarlas y así obtener el valor deseado.

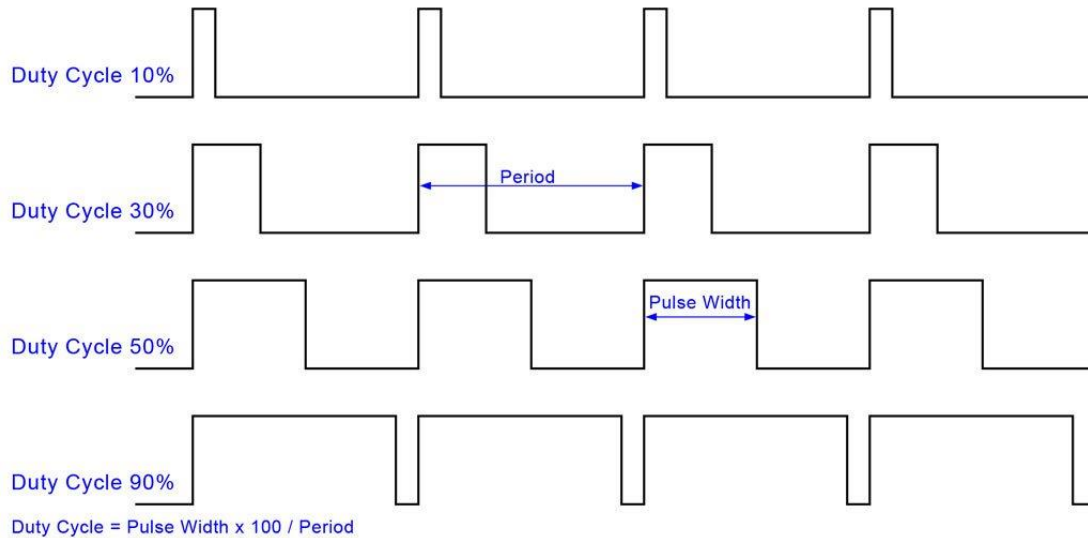
Pero para poder comprender como el sistema consigue regular la velocidad del motor, primero necesitamos entender la tecnología PWM del Arduino:

Las Salidas PWM (Pulse Width Modulation) permiten generar salidas analógicas desde pines digitales. Arduino Uno no posee salidas analógicas puras.

La modulación por ancho de pulsos de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica, ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Duty cycle = (tiempo que la salida está a uno o HIGH)/ (periodo de la función)





En Arduino la frecuencia de PWM es de 500Hz, o lo que es lo mismo, $2 \cdot 10^{-3}$ s

Como nos encontramos en corriente continua, lo que nos encontramos es el valor medio de la función:

$$V_{MED} = \frac{1}{T} \cdot \int_0^T V(T)$$

Como trabajamos con 9V, una onda de 50% de ciclo de trabajo ($1 \cdot 10^{-3}$ s activa y $1 \cdot 10^{-3}$ s a 0) se recibirá como una tensión continua de 4.5V.

Pese a sufrir cortes de tensión, el motor no se para debido a que es un sistema mecánico que mantiene una inercia. Si los periodos sin corriente fueran muy prolongados, se acabaría parando por esa misma inercia que tiene.

Por lo que la tensión proporcionada en realidad será:

$$V_T(t) = 9 \cdot k_d(t)$$

Donde $k_d(t)$ es la constante de tiempo de trabajo, y varía según la tensión que queramos obtener.

Para terminar de obtener las ecuaciones del sistema, obtener la función de transferencia y obtener el diagrama de bloques, primero tendremos que entender del todo el sistema del control:

1) Se introduce el número de revoluciones deseadas



- 2) Con estas revoluciones el sistema obtiene el porcentaje de tiempo de trabajo
- 3) Con el porcentaje de tiempo de trabajo
- 4) Sabiendo el tiempo de trabajo tendremos la cantidad de tensión y corriente que le llega al motor
- 5) Teniendo estos dos parámetros, podemos calcular la tensión que cae en el rotor.
- 6) Con la tensión que cae en el rotor calculamos las revoluciones del eje.
- 7) El tacómetro óptico lee las revoluciones del eje, y las convierte a una señal cuadrada
- 8) Pasa esta señal al porcentaje de la señal máxima, para poder realizar la comparación con el porcentaje de tiempo de trabajo
- 8) Realimentamos el sistema con esta señal, comparándolo con la del motor, lo que modifica el tiempo de trabajo según necesitemos aumentar o disminuir los parámetros del motor

Con todo esto las ecuaciones del sistema quedan:

$$k_{\text{tiempo de trabajo desada}}(t) = \frac{1}{w_{\text{maxima}}} \cdot w_{\text{deseada}}(t)$$

$$k_{\text{tiempo de trabajo motor}}(t) = 2 \cdot k_{\text{tiempo de trabajo desada}}(t) - k_{\text{tiempo tacómetro}}(t)$$

$$U_{\text{motor}}(t) = k_{\text{tiempo de trabajo motor}}(t) \cdot (V_{CC} - 2 \cdot V_{CE})$$

$$i(t) = k_{\text{tiempo de trabajo motor}}(t) \cdot I_{\text{maxima}}$$

$$U_M(t) = i(t) \cdot R_{\text{motor}} + L \cdot \frac{d[i(t)]}{T} + e_b(t)$$

$$e_b(t) = k_b \cdot w_{\text{motor}}(t)$$

$$U_{\text{tacómetro}}(t) = k_{\text{tacométrica}} \cdot w_{\text{motor}}(t)$$

$$U_{\text{tacómetro}}(t) = k_{\text{tiempo tacómetro}}(t) \cdot V_{\text{máxima del tacómetro}}$$

Realizamos las trasformadas de Laplace para poder realizar el diagrama de bloques:

$$k_{\text{tiempo de trabajo desada}}(s) = \frac{1}{w_{\text{maxima}}} \cdot w_{\text{deseada}}(s)$$

$$k_{\text{tiempo de trabajo motor}}(s) = 2 \cdot k_{\text{tiempo de trabajo desada}}(s) - k_{\text{tiempo tacómetro}}(s)$$

$$U_{\text{motor}}(s) = k_{\text{tiempo de trabajo motor}}(s) \cdot (V_{CC} - 2 \cdot V_{CE})$$

$$i(s) = k_{\text{tiempo de trabajo motor}}(s) \cdot I_{\text{maxima}}$$

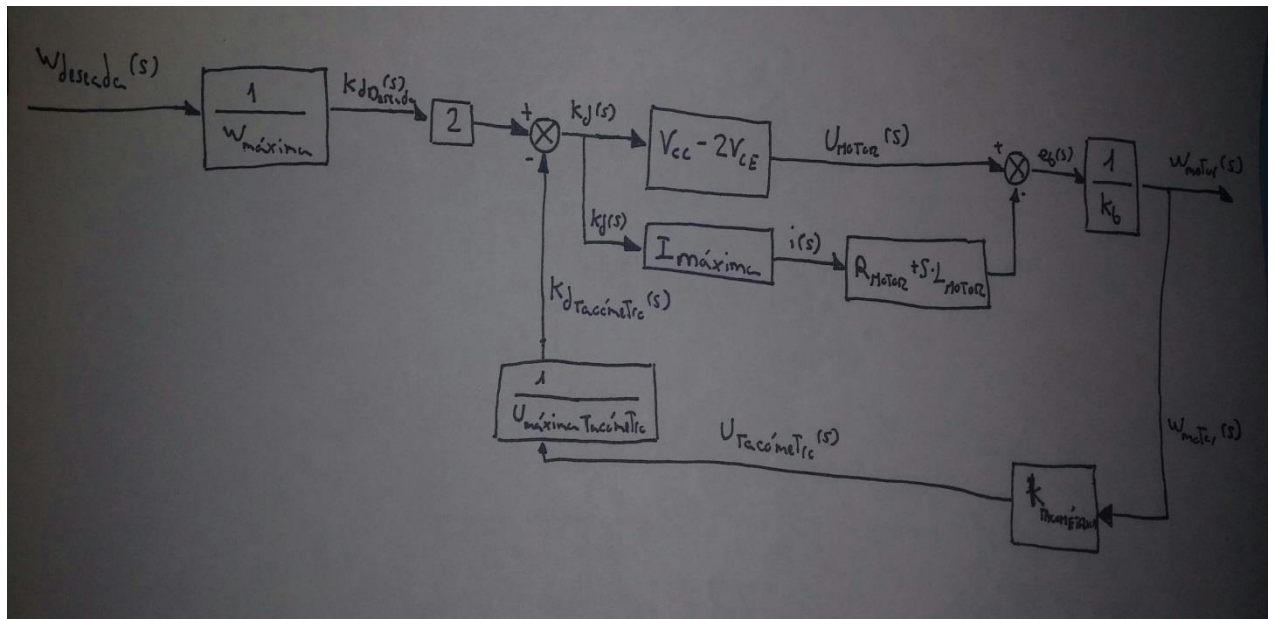
$$U_M(s) = i(s) \cdot R_{\text{motor}} + L \cdot S \cdot i(s) + e_b(s)$$

$$e_b(s) = k_b \cdot w_{\text{motor}}(s)$$

$$U_{\text{tacómetro}}(s) = k_{\text{tacométrica}} \cdot w_{\text{motor}}(s)$$

$$U_{\text{tacómetro}}(s) = k_{\text{tiempo tacómetro}}(s) \cdot V_{\text{máxima del tacómetro}}$$

El diagrama de bloques queda:



Con este diagrama la función de transferencia del sistema nos quedaría:

$$M(s) = \frac{w_{\text{motor}}(s)}{w_{\text{deseada}}(s)}$$

$$M(s) = \left[\frac{1}{w_{\text{máxima}}} \cdot 2 \right] \cdot \left[\frac{[(V_{CC} - 2V_{CE}) + [I_{\text{máxima}} \cdot (R_{\text{motor}} + S \cdot L_{\text{motor}})]] \cdot \frac{1}{k_b}}{1 + \left[[(V_{CC} - 2V_{CE}) + [I_{\text{máxima}} \cdot (R_{\text{motor}} + S \cdot L_{\text{motor}})]] \cdot \frac{1}{k_b} \right] \cdot \left[k_{\text{tacométrica}} \cdot \frac{1}{V_{\text{máxima tacómetro}}} \right]} \right]$$



Los parámetros constantes de nuestro circuito son:

La tensión colector emisor de los transistores será la que tengan en saturación estos. Esto se debe a que, como la salida PWM está pasándolos todo el rato entre corte y saturación, la única tensión que nosotros vamos a percibir cuando esté activa la salida serán los 0.2V de saturación.

$$V_{CC} = 9V$$

$$\omega_{m\acute{a}xima} = 10000 \text{ rpm}$$

$$I_{m\acute{a}xima} = 0.6A$$

$$R_{motor} = 20\Omega$$

$$L_{motor} = 22 \cdot 10^{-3}H$$

$$k_b = 1.2 \cdot 10^{-3} \frac{V}{rpm}$$

$$k_{tacometr\acute{a}ica} = 1.05 \cdot 10^{-3} \frac{V}{rpm}$$

$$V_{m\acute{a}xima \text{ tac\acute{o}metro}} = 10.5V$$

Con estos datos la f.d.t. del sistema queda:

$$M(s) = \left[\frac{1}{10000} \cdot 2 \right] \cdot \left[\frac{[(8.6) + [0.6 \cdot (20 + S \cdot 22 \cdot 10^{-3})]] \cdot \frac{1}{1.2 \cdot 10^{-3}}}{1 + \left[\left[[(8.6) + [0.6 \cdot (20 + S \cdot 22 \cdot 10^{-3})]] \cdot \frac{1}{1.2 \cdot 10^{-3}} \right] \cdot \left[1.05 \cdot 10^{-3} \cdot \frac{1}{10.5} \right] \right]} \right]$$

$$M(s) = 2 \cdot 10^{-4} \cdot \left[\frac{23833 + 11 \cdot S}{1 + [2.238 + 11 \cdot 10^{-4} \cdot S]} \right] = \frac{4.767 + 2.2 \cdot 10^{-3} \cdot S}{3.238 + 1.1 \cdot 10^{-3} \cdot S}$$

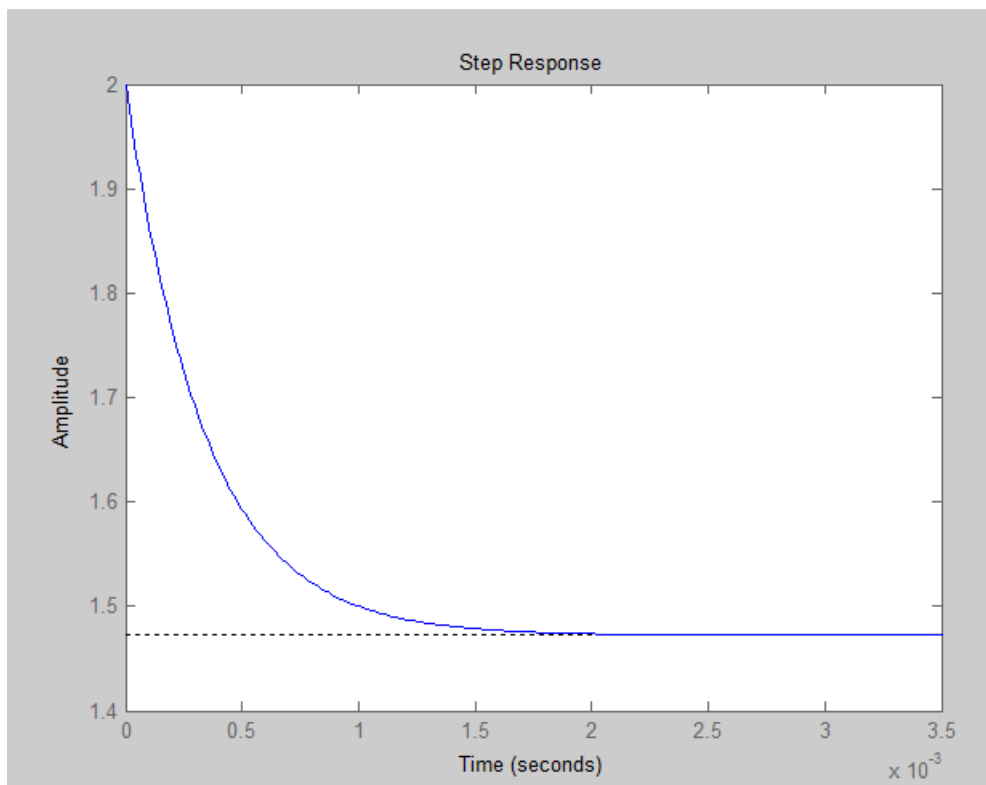
$$M(s) = \frac{4.767 + 2.2 \cdot 10^{-3} \cdot S}{3.238 + 1.1 \cdot 10^{-3} \cdot S}$$



Matlab

Pasamos a realizar las simulaciones de nuestra f.d.t en Matlab:

```
>> g1=tf([2.2e-3 4.767],[1.1e-3 3.238])  
  
g1 =  
  
    0.0022 s + 4.767  
    -----  
    0.0011 s + 3.238  
  
Continuous-time transfer function.  
  
>> step(g1)  
>>
```



Sabemos que en realidad nuestra función de transferencia no es así, ya que muchos de los parámetros que hemos usado para el motor han sido obtenidos mediante suposiciones y ajustes al no ser un motor normalizado.

Montaje

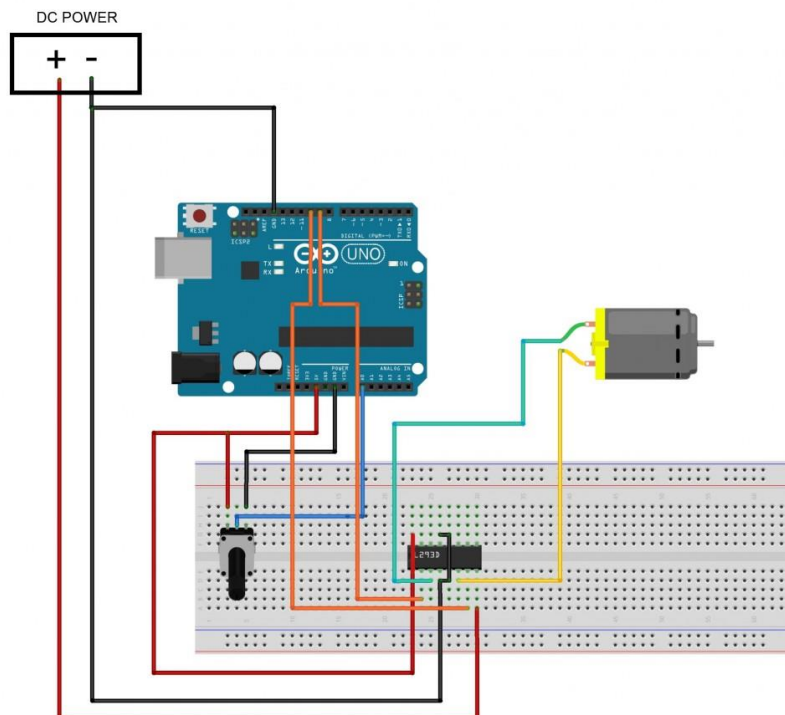
Para el montaje primero hicimos funcionar el circuito del puente en h, siguiendo varias etapas:

-La primera etapa se basó en hacer funcionar el control del sentido de giro del circuito compuesto por motor e integrado 1923d.

Para ello conectamos el integrado a dos pines digitales del Arduino, con lo cual solo teníamos la opción de hacer funcionar el motor al máximo de revoluciones en un sentido o en otro, o de pararlo. Una vez comprobado esto, pasamos a la segunda etapa.

-En la segunda etapa nos centramos en el control de velocidad.

Para ello incluimos en el circuito un potenciómetro, y conectamos el integrado a dos pines PWM. Con esto ya teníamos control sobre el sentido de giro y la velocidad.



-Antes de pasar el control de velocidad al serial, iniciamos el montaje del tacómetro:

Conectamos un diodo Led y un fotodiodo para que haga de receptor, conectándolo a un pin de entrada analógico. Como el fotodiodo recibe mucha luz del exterior, canalizamos ambos para que la única luz que reciba el fotodiodo provenga del Led, y le sea más fácil de cortar al aspa.

Los pines de entrada analógica presentan un “ruido” base, el cual se incrementa con el uso de cables, por lo cual tenemos que contar con ellos en el programa, aunque siempre incluirán un fallo. También se podría haber mitigado este ruido con la conexión de un condensador en paralelo con la entrada, actuando como filtro paso bajo.

Una vez juntados ambos circuitos, y comprobado que las lecturas del tacómetro son coherentes con la potencia elegida en el potenciómetro, pasamos a la tercera etapa.

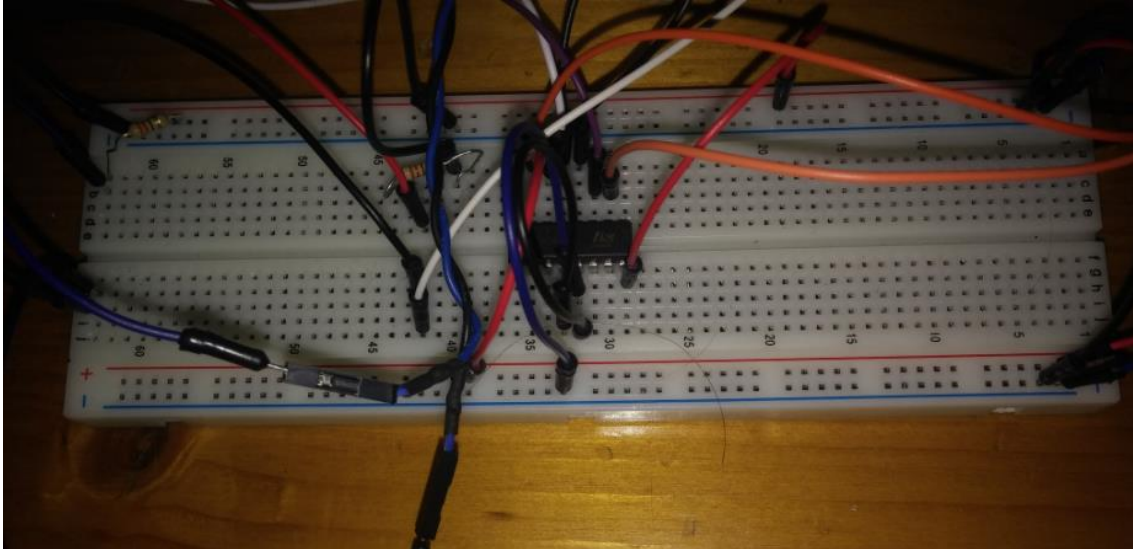
-En la última etapa eliminamos el elemento potenciómetro, realizando ahora la regulación de velocidad por la lectura del tacómetro, y por la introducción de datos a través del monitor serie.

Si la tensión del tacómetro es mayor que la introducida en el serial, disminuye la salida para tener menos revoluciones, si es mayor, la aumenta.

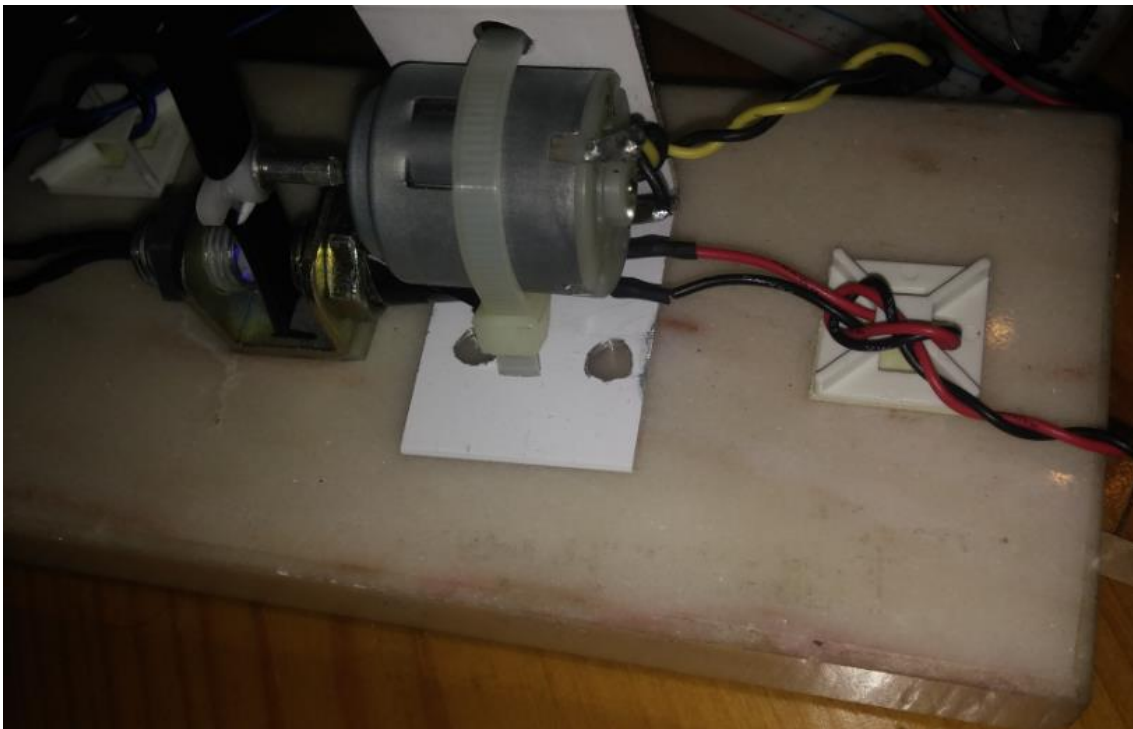
➔Fuente de alimentación externa 9V:



→ Circuito electrónico:



Circuito mecánico:





Programa

```
int ord_eje=0;
```

```
int ord_rpm=0;
```

```
int ord_pot=0;
```

```
int val_luz_act=0;
```

```
int val_luz_ant=0;
```

```
long last_p=0;
```

```
int cont_p=0;
```

```
int mseg_p=1000;          // tiempo de impresión
```

```
long last_x=0;
```

```
int cont_x=0;
```

```
int mseg_x=100;          // tiempo de ejecución
```

```
int nvez=0;
```

```
long sval=0;
```

```
int sens=5; //Valor umbral de la distancia entre valores de lectura de luz y de oscuridad
```

```
int nPalas=2; //número de palas
```

```
//-----
```



```
void setup() {  
    Serial.begin(9600);  
  
    pinMode(6,OUTPUT); //declaramos los pines como salidas  
  
    pinMode(5,OUTPUT);  
  
    pinMode(13,OUTPUT);  
}  
  
//-----  
  
void loop()  
{  
    val_luz_act=0;  
    val_luz_act=analogRead(A0);  
  
    nvez++;  
    sval= sval+val_luz_act;  
  
    if(abs(val_luz_act-val_luz_ant)>sens)  
    {  
        cont_p++;  
        cont_x++;  
    }  
    val_luz_ant=val_luz_act;  
  
    // -----
```



```
if(millis()-last_p>=mseg_p)
{
    if (Serial.available(>0)
    {
        LEER_ORD();
    }
}

// -----

if(millis()-last_x>=mseg_x)
{
    double rps=((double)cont_x/(nPalas*2))*(1000.0/mseg_x);
    double rpm=((double)cont_x/(nPalas*2))*(60000.0/mseg_x);

    if (rps>ord_rpm)
        ord_pot=(ord_pot*95)/100;
    if (rps<ord_rpm)
        ord_pot=(ord_pot*105)/100;

    if (ord_pot>300)
        ord_pot=300;
    if (ord_pot<0)
        ord_pot=ord_rpm/4;
```



```
EJEC_ORD();

cont_x=0;

last_x=millis();

}

// -----

if(millis()-last_p>=mseg_p)

{

    double rps=((double)cont_p/(nPalas*2))*(1000.0/mseg_p);

    double rpm=((double)cont_p/(nPalas*2))*(60000.0/mseg_p);


    Serial.print(" ");

    Serial.print (ord_eje);

    Serial.print(" X ");

    Serial.print (ord_rpm);

    Serial.print(" R ");

    Serial.print (ord_pot);

    Serial.print(" P ");


    Serial.print(nvez);

    Serial.print(" V ");

    Serial.print((sval/nvez));
```



```
Serial.print(" M ");
```

```
Serial.print(cont_p);
```

```
Serial.print(" C RPS ");
```

```
Serial.print(rps);
```

```
Serial.print(" RPM ");
```

```
Serial.print(rpm);
```

```
Serial.print(" VAL ");
```

```
Serial.println(val_luz_act);
```

```
nvez=0;
```

```
sval=0;
```

```
cont_p=0;
```

```
last_p=millis();
```

```
}
```

```
}
```

```
//-----
```



```
void LEER_ORD(void)
{
    int ord_num=0;
    while (Serial.available()>0)
    {
        char ord_ent=Serial.read();
        if (ord_num<200)           // limite RPM 200x10=2000
            if (ord_ent>='0' && ord_ent<='9')
                ord_num=(ord_num*10)+int(ord_ent-48);
    }

    switch(ord_num)
    {
        case 0:    //PARADA
        {
            ord_rpm=0;
            ord_eje=ord_num;
            ord_pot=0;
            break;
        }

        case 1:    //ARRANQUE IZQUIERDA
        {
            ord_eje=ord_num;
```




```
        break;
    }

    case 3:        //ARRANQUE DERECHA
    {
        ord_eje=ord_num;
        break;
    }

    default:      //PONER VELOCIDAD
    {
        ord_rpm=ord_num;
        ord_pot=ord_rpm/4;
    }
}

//-----

void EJEC_ORD(void)
{
    switch(ord_eje)
    {
```



```
case 0:    //PARADA
{
    analogWrite(6,0);
    analogWrite(5,0);
    break;
}

case 1:    //IZQUIERDA
{
    analogWrite(6,0);
    analogWrite(5, ord_pot);
    break;
}

case 3:    //DERECHA
{

    analogWrite(5,0);
    analogWrite(6, ord_pot);
    break;
}

}

}
```



Observaciones

Este proyecto ha sido de gran interés, ya que presenta problemas de diversas áreas.

Me ha permitido entender en profundidad el funcionamiento de la tecnología PWM, gracias a la ayuda de diversos profesores de la escuela.

El primer reto que se me planteó fue el estudio de los transistores que componen el circuito. Tras conseguir realizar el montaje del circuito “puente en h” por componentes, decidí emplear un circuito integrado (1293d) con para evitar ruidos y reducir el error de las lecturas.

El segundo problema que surgió fue el hecho de elegir un tacómetro adecuado, ya que nuestro motor venía solo con un palo de acero conectado al rotor. Me decanté por el tacómetro óptico ya que era el más sencillo de construir, y no requería de más circuitos integrados (también podría haber recurrido a un tacómetro magnético). El siguiente paso fue instalar un aspa al rotor, para que pudiera cortar el haz de luz del emisor al receptor en el tacómetro.

Una vez elegido el tipo de tacómetro, me decidí por un receptor basado en un fotodiodo en vez de un fototransistor, ya que estos últimos no tienen tanta rapidez de lectura, y aumentaba bastante el ruido del pin analógico.

Pese a todo el sistema seguía teniendo un error producido por el ruido de entrada en la lectura analógica, por lo que tuvimos que realizar modificaciones en el programa para ajustarlo y tratar de solventarlo lo máximo posible.

El circuito presenta un error bastante perceptible debido en primer lugar a que no tenemos un motor normalizado, por lo que muchos de sus parámetros han sido obtenidos mediante medidas, suposiciones y medias con parámetros de motores normalizados. A esto hay que sumarle el error de lectura.

Debido al error en los parámetros también la función de transferencia del circuito presenta fallos, ya que sabemos que es un sistema subamortiguado, debido a que las revoluciones van a ir oscilando cada vez menos alrededor de una oscilación deseada, hasta que llegue el momento en que la alcancen