

Power Electronics for a trackerbot

Students: Luis Muñiz Pasarín , Sergio Llano Martínez

Advisers: Manuel Rico-Secades , Antonio Javier Calleja Rodriguez

Work Reference: WED-2013-04

WRE - Workroom on Renewable Energy

Gijón Polytechnic Campus of Engineering (EPI Gijón)

Abstract

Today, robotics is a topic of growing interest is starting to increase its presence in our lives and in the industry. There are different types of robots: cooking, assembly, cleaning, competition ...; the latter are the focus of our work, namely the crawler robot. Work crawler robot is to follow lines that form a maze and try to reach as quickly as possible. To achieve this requires a series of robot components: sensors to follow lines, two engines, a PIC, a control program, an H-bridge and its drivers.

Objetivos

Hoy en día la robótica es un tema de creciente interés, está comenzando a incrementar su presencia en nuestras vidas y en la industria. Existen robots de diferentes tipos: de cocina, de ensamblaje, de limpieza, de competición...; Estos últimos son el centro de nuestro trabajo, en concreto el robot rastreador.

La labor del robot rastreador es seguir líneas que forman un laberinto e intentar llegar lo más rápido posible. Para conseguirlo, este robot necesita una serie de componentes principales: Sensores para seguir las líneas, dos motores, un PIC, un programa de control, un puente en H y sus drivers.

Keywords

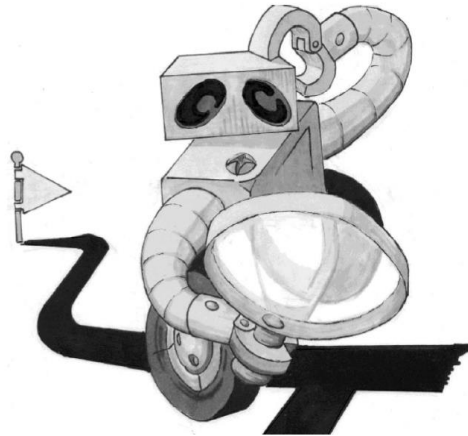


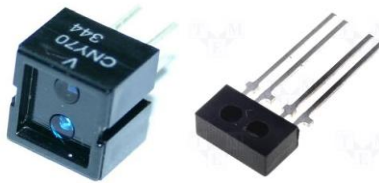
Figura 1.- Robot rastreador

Para alimentar todo este sistema hemos elegido unas baterías que conectadas nos dan 7.2 V, suficiente voltaje para un buen funcionamiento del robot.

El PIC que vamos a utilizar, el 16F690, necesita una alimentación de 5 V. Para reducir la tensión a este valor, es necesario utilizar un regulador de tensión, que puede realizarse a través de un circuito de resistencias o utilizando un integrado, como por ejemplo el **MC7805ABD2TG**, que tiene un precio de 0.81 €.

(disponible en el enlace <http://docs-europe.electrocomponents.com/webdocs/0609/0900766b806090e4.pdf>).

El seguimiento de las líneas se realizará mediante sensores blanco/negro, que se basan en un fotodiodo y un fototransistor. Su funcionamiento consiste en la detección de la luz emitida por el fotodiodo, que es reflejada en la superficie y captada por el fototransistor, que da un voltaje en función de la luz recibida. Barajamos la idea de utilizar los **TRCT1000** (precio alrededor de 1 €) o los **CNY70**(con un precio ligeramente más caro, 1.15 €). Los primeros tienen un encapsulado más pequeño y ligero, pero los segundos están mejor protegidos de la luz externa, por lo que darán una medición más fiable.



Estos componentes pueden encontrarse en las páginas siguientes: <http://www.vishay.com/docs/83751/cny70.pdf>
<http://www.vishay.com/docs/83752/tcrt1000.pdf>

referencias: <http://www.youtube.com/watch?v=VrbrjHJS6Nw>
<http://www.youtube.com/watch?v=Dk1CLYCKvTM>

I. MOTOR

Una vez conseguido un buen seguimiento de líneas, nuestro objetivo ahora será conseguir el robot más rápido. En la elección del motor será importante tener en cuenta la velocidad a la que queremos ir, el peso del robot y el precio.

En nuestro caso estimamos que el robot final tendrá un peso de unos 100 gramos, y queremos que vaya a una velocidad máxima de aproximadamente 1 m/s. Teniendo en cuenta que las ruedas tendrán un radio aproximado de 1.5 cm, podemos calcular la velocidad que debe tener el motor y el par necesario.

Los cálculos nos dejan una velocidad de 636.6 rpm y un par necesario de 3 mNm.

Utilizaremos un motor con escobillas, de corriente continua. Las ventajas de los motores de corriente continua incluyen un excelente control de la velocidad y la capacidad de proporcionar alto par a bajas velocidades. Hemos elegido un motor de maxonmotor; el modelo **DCX 10 L brushed DC motor Ø10 mm**, con escobillas de metal precioso y cojinete sinterizado. Tiene un peso de 10.4 gramos, un par de 2.22 mNm y una velocidad nominal de 7100 rpm. Como necesitamos un par mayor y una velocidad menor, le añadiremos una etapa reductora.

Todas las reductoras de maxon motor son compatibles con sus motores. Utilizaremos el modelo **Planetary Gearhead GP 10 K Ø10 mm 110308** con la cual reduciremos la velocidad a 1775 rpm y aumentaremos el par a 8.88 mNm, suficiente para mover el robot.

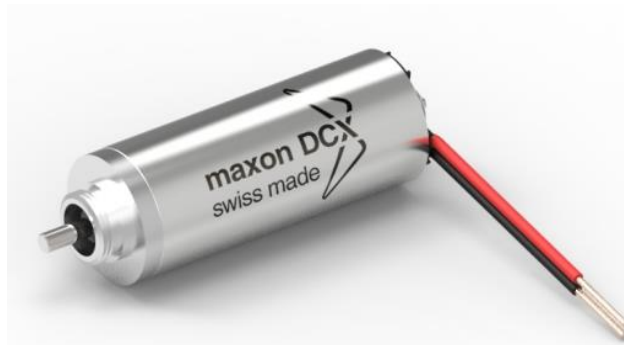
Este motor es caro, su precio ronda los 70 €, pero lo hemos escogido porque maxonmotor nos da todos sus parámetros, lo que nos facilita la simulación.

Cuando se realice el prototipo real será importante el precio, por lo que escogeríamos un motor más barato, como por ejemplo el **35:1 Metal Gearmotor 15.5Dx30L mm**, de Pololu, cuyo precio ronda los 20 €.

Referencias:

<http://www.maxonmotor.es/maxon/view/catalog/>

<http://www.treffer.com.br/produtos/maxon/redutores/pdf/214.pdf>



II. PUENTE EN H

El puente en H es el circuito más utilizado para controlar el sentido de giro de un motor. Adquiere dicho nombre por la localización de los transistores en el circuito. El esquema inicial es sencillo: Se parte del motor en la parte central y queda rodeado por 4 interruptores. Activando el superior del lado izquierdo y el inferior del lado derecho conseguiremos que gire en un sentido, mientras que si activamos los contrarios haremos que gire en el otro sentido. Es importante tener en cuenta que nunca deben activarse los dos transistores de la misma rama a la vez, ya que provocaríamos que toda la corriente de la fuente de alimentación pase a través de ellos pudiendo destruirlos.

Los transistores utilizados pueden ser bipolares o unipolares. Los transistores Mosfet poseen una serie de ventajas respecto a los bipolares:

- Consumo en modo estático muy bajo.
- Funcionamiento por tensión, son controlados por voltaje por lo que tienen una impedancia de entrada muy alta. La intensidad que circula por la puerta es del orden de los nanoamperios.
- La velocidad de conmutación es muy alta, siendo del orden de los nanosegundos.

Nosotros hemos optado por usar transistores Mosfet, en concreto el modelo **IRFD014**(precio alrededor de 1 €).

Datasheet:

<http://www.vishay.com/docs/91125/sihfd014.pdf>

En este circuito nosotros trabajamos con tensiones pequeñas, aun así siempre es necesario proteger. Este modelo aguanta hasta 60 V y tiene incluido el diodo de protección, con lo que estará protegido frente a los picos de tensión inversa producidos por el arranque del motor y por el cambio de sentido.

Las pérdidas en conmutación y en conducción de los mosfet se calculan de la siguiente manera:

Pérdidas de conmutación

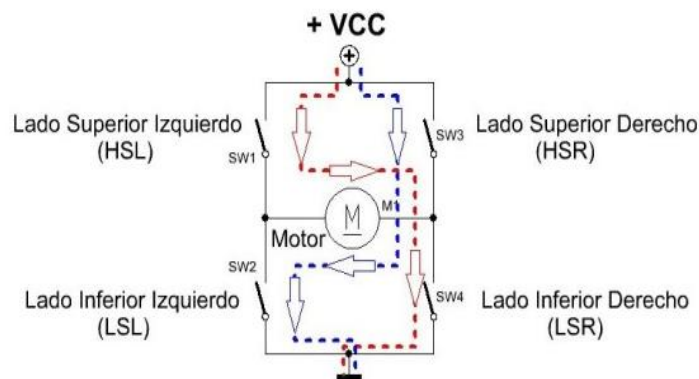
$$\frac{1}{2} * V_s * I_s * f_s * (t_{on} + t_{off}) = 0.5 * 7.2 * 0.649 * 20 \text{kHz} * (10 + 13 \text{ ns}) = \mathbf{1.07E-3 \text{ W}}$$

Pérdidas en conducción

$$R_{ds} * (I_{rms})^2 = 0.20 * 0.210 = \mathbf{0.042 \text{ W}}$$

Como se puede observar las pérdidas son insignificantes y no supondrán ningún problema.

Colocaremos en serie con el motor una resistencia de 0.25Ω , formada con 4 resistencias de 1Ω en paralelo, para la protección frente a sobrecorrientes. La potencia disipada será $P = VI = RI^2 = 0.25 * 0.649^2 = \mathbf{0.105 \text{ W}}$, que no supondrá problema alguno, ya que cada una de las resistencias de 1Ω soporta 0.5 W .



III. DRIVERS

Para controlar la activación de los Mosfets del puente en H será necesario el uso de Drivers. Estos pueden tener incluidos los mosfets o venir sin ellos, en nuestro caso como ya disponemos de los Mosfet mencionados anteriormente, hemos conseguido un driver que no los tiene incluidos. Los driver pueden hacerse manualmente componente a componente o utilizar uno integrado, nosotros tenemos el **IR2110** (precio en torno a 1.5 €), que es integrado.

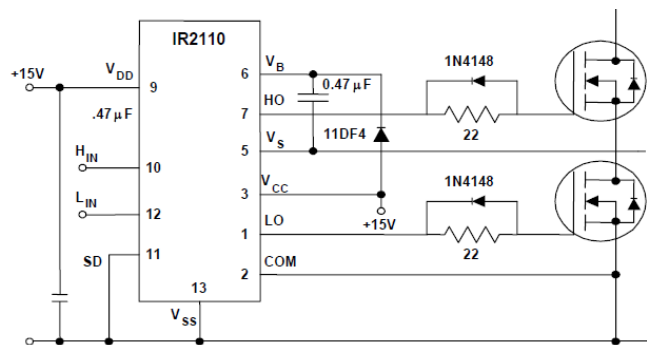
<http://www.irf.com/product-info/datasheets/data/ir2110.pdf>

Con este driver podemos controlar dos mosfets, ya que dispone de dos salidas, HO y LO (high output y low output). La alimentación de este driver se realiza por independiente utilizando una fuente de 15 V, en su datasheet se puede ver que la alimentación necesaria es de 10-20V. El circuito que viene dentro de este integrado se conoce por el nombre de bootstrap, el cual se podría construir manualmente, utilizando un optoacoplador por cada transistor Mosfet, un condensador o dos (dependiendo de si la salida del PIC está al mismo nivel de tensión que llega al motor), que servirán para mantener cargado el Mosfet, y un par de diodos por seguridad.

Si intentáramos construirlo por nuestra cuenta, el precio final sería algo mayor al de los integrados, por lo cual es más rentable utilizar estos últimos.

Para controlar este driver que hemos escogido, hay que mandar una señal por la salida LO para cargar los dos condensadores que servirán para controlar los mosfet. Una vez hecho esto, para un correcto funcionamiento, se deben de mandar pulsos a alta frecuencia, controlados por el PWM, por la salida HO, para evitar que el condensador se descargue, lo cual haría que el circuito dejara de funcionar hasta que se le metiera otro pulso por LO.

Los componentes recomendados en la application note de este driver son los siguientes:

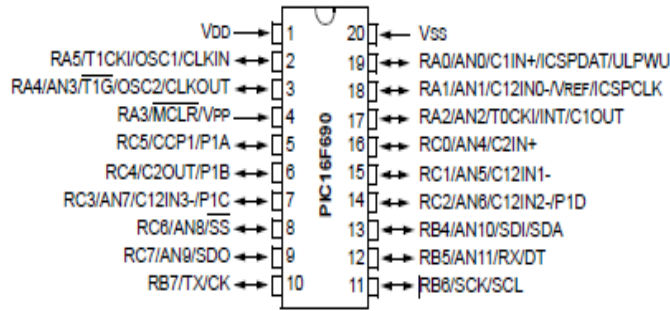


V.PIC

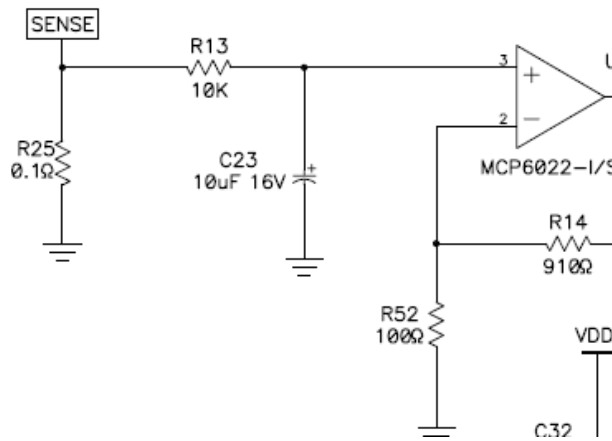
El PIC que hemos elegido es el 16f690, que se puede encontrar en el siguiente enlace:

<http://ww1.microchip.com/downloads/en/devicedoc/41262a.pdf>

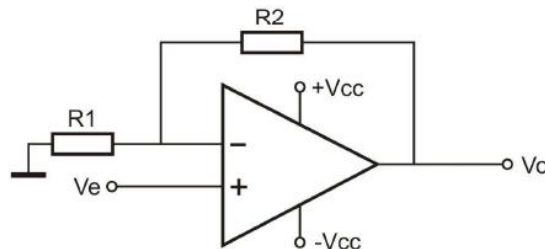
Este pic tiene las siguientes conexiones:



Las patas VDD y VSS irán conectadas a 5V y a 0V respectivamente. Para la señal PWM se usarán las patas P1A,P1B,P1C,P1D tal y como nos indica el datasheet del pic. Utilizaremos la pata RB4 para conectar un pulsador de inicio de marcha y el AN0 para conectar un sensor, que medirá la tensión que pasa por la resistencia en serie con el motor, de tal forma que si el voltaje por ella es superior a un determinado valor pararemos el programa, para evitar que los components sufran daño alguno. El sensor sigue el siguiente esquema:



Como queremos proteger al doble de la corriente nominal, tendremos la siguiente tensión $0.25 \cdot 1.3 = 0.3 \text{ V}$, que como es un valor muy bajo será necesario amplificar, para ello usaremos un montaje no inversor, siguiendo el modelo mostrado en la figura:



Amplificaremos hasta 4.5V por lo que necesitamos una $G=15$ ($R1=1 \text{ K}$ $R2=14\text{K}$). La red RC inicial se utiliza para estabilizar la tensión. La parada del robot se realizaría mediante software utilizando el ADCON del Pic.

VI.PWM

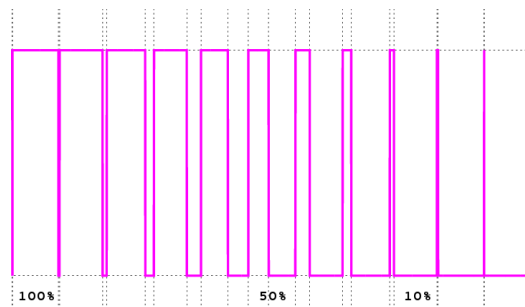
La modulación por ancho de pulsos es una técnica utilizada para el control de la velocidad de motores, en nuestro caso de corriente continua. Recurrimos a este método debido a que no requiere gran cantidad de componentes y a su flexibilidad para variar la velocidad en funcionamiento.

Es importante revisar la hoja de datos de cada PIC para identificar varios puntos que serán relevantes a la hora de configurar el PWM. Algunos de estos puntos son:

- Timer con el que opera el módulo.
- Resolución del módulo (cantidad de bits).
- Frecuencia de trabajo.
- Pin del módulo CCP.

Para establecer el duty debemos saber la resolución de nuestro PWM , ya que si es de 10 bits , el tiempo máximo que nuestra señal puede estar en uno es 1024 , mientras que si la resolución es de 8 bits será 255.

Debe tenerse en cuenta que la frecuencia que pongamos debe ser lo suficientemente elevada para observar una marcha estable, pero debe ser inferior a la frecuencia de conmutación de los MOSFET. Además es conveniente que esté por encima de 20 Khz para evitar el ruido de la conmutación. Dependiendo de la frecuencia que utilicemos puede que el motor gire demasiado rápido y necesitemos una velocidad menor, para regular esto se usa el PWM.



El 16f690 tiene modulo EPWM, que dispone de 4 salidas PWM de resolución 10 bits. El timer con el que opera el PWM en el 16f690 es el TMR2. El datasheet nos da las formulas necesarias para calcular el duty:

PWM period:

$$PWM\ period = [(PR2) + 1] \cdot 4 \cdot TOSC \cdot (TMR2\ prescale\ value)$$

PWM duty cycle time:

$$PWM\ duty\ cycle = (CCPR1L:CCP1CON<5:4>) \cdot TOSC \cdot (TMR2\ prescale\ value)$$

Max.PWM resolution per frequency:

$$Resolution = \frac{\log\left(\frac{FOSC}{FPWM \cdot TMR2\ Prescaler}\right)}{\log(2)} \text{ bits}$$

El duty hará variar el ancho del impulso de conexión para obtener así un funcionamiento controlado desde la detención absoluta hasta la marcha a máxima potencia.

De tal forma que por ejemplo, para un PWM de resolución 8 bits(255 valores), si le asignamos a la variable del programa que representa el duty un cero, el motor estará parado, de igual manera que para valores muy pequeños, ya que la onda no tendrá el ancho suficiente como para dar al motor la energía necesaria para comenzar a girar. Pero si incrementamos la variable del duty a valores superiores, por ejemplo el valor 120, tendremos $120/255 \cdot 100 = 47\%$, por lo que conseguiremos hacer girar al motor a una velocidad de aproximadamente el 47% de la velocidad máxima.

Existe un registro de auto-shutdown (ECCP), mediante el cual podemos detener el PWM cuando se detecta un determinado evento, por ejemplo un exceso de tensión, de tal forma que también nos puede servir a modo de protección. Este módulo puede ser configurado para auto-restarts, para ello debemos poner a uno el bit PRSEN del registro PWM1CON. Estas opciones están más detalladas en las páginas 129-130 del datasheet del Pic16F690: (<http://ww1.microchip.com/downloads/en/devicedoc/41262a.pdf>)

VII. PSIM

Para observar cómo sería la respuesta de este sistema, hemos montado un puente en H en Psim, tal y como se presenta en la gráfica 1. Como ya sabemos, el motor da un pico de corriente en el arranque, como se puede observar en la gráfica 2. El que la línea no sea fina y continua se debe al PWM, que va abriendo y cerrando el circuito, aunque de forma muy rápida, como vemos en la gráfica 3, que es el resultado de hacer zoom en la gráfica 2. Simulando la tensión del motor podemos ver como queda regulada por el PWM, y llega hasta la tensión con la que alimentamos el puente, en nuestro caso 7.2V (gráfica 4).

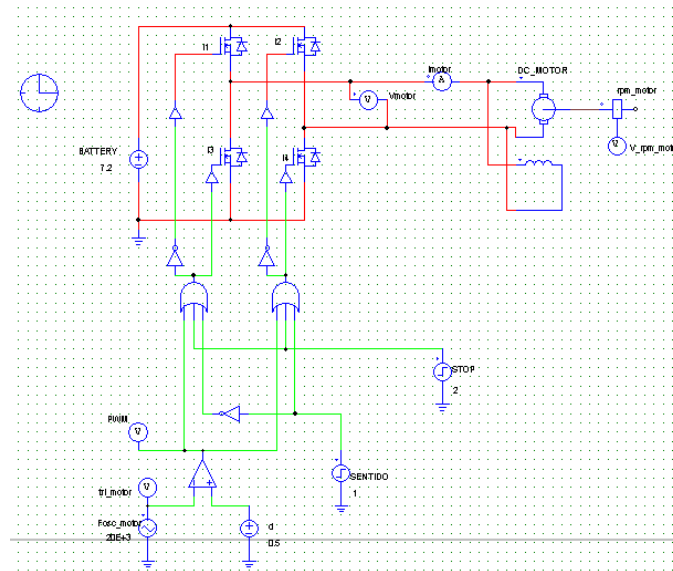
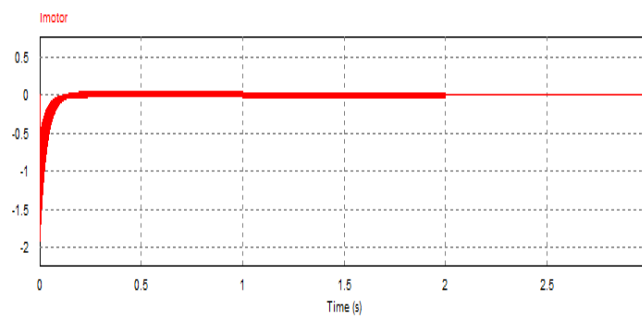
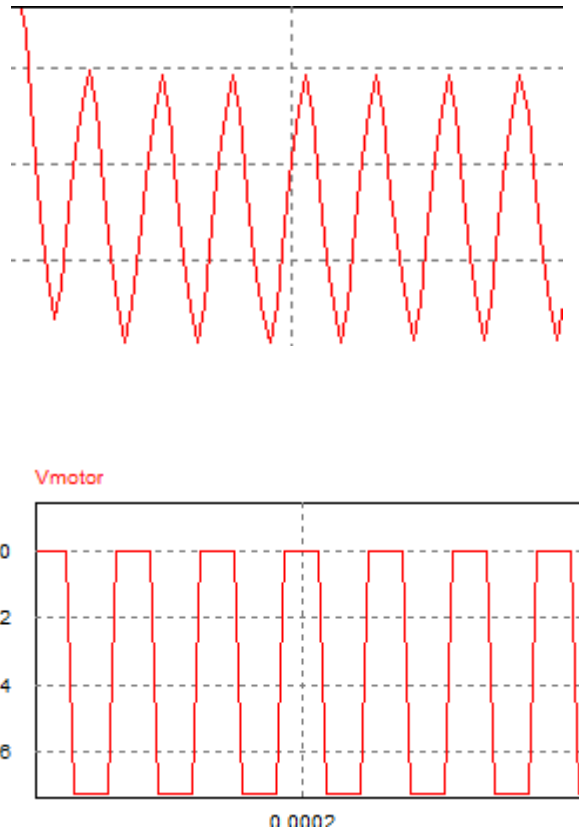


Figura 8.- Simulación del Puente en H para selección de componentes





VIII.PCB Design

Es de interés tener una idea de como quedaría una placa PCB con este montaje, para ello realizamos el montaje del puente en H completo, con el PIC y los drivers. Para hacer las conexiones utilizamos la función “TO”, que nos permite conectar varios puntos (misma net) sin ir tirando cables.

Una vez realizado el esquema vamos a tool/translate to PCB y tras una serie de pasos podemos realizar las conexiones en la placa PCB. Además el programa ofrece la posibilidad de tener una visión en 3D del resultado final.

