



MADRIDBOT´2007

Calyptratus

Marián de Miguel Pérez – Roberto González Felipe

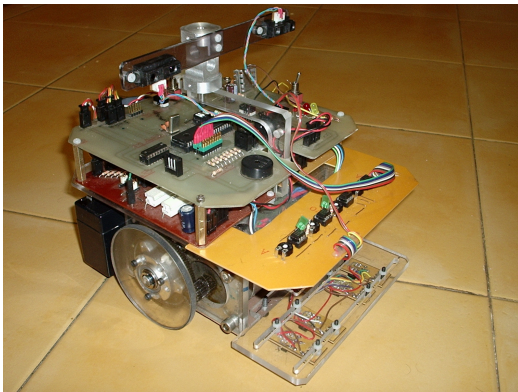
mariaandemiguels@hotmail.com – robergrajal@yahoo.es

Robótica Miranda

Centro Específico de Formación Profesional de Palencia

Resumen

Nuestro robot fue diseñado para la participación en la prueba de Laberinto de Hispabot de 2006. Al modificarse en demasía la normativa (con respecto a la del año 2004), nuestro robot quedó inservible hasta que se nos ocurrió sacar provecho a nuestro trabajo acoplándole unos nuevos sensores para que pudiera participar en la prueba de rastreadores. Ahora que nuestro robot es compatible tanto con la prueba de laberinto como con la prueba de rastreadores hemos decidido llamarlo *Calyptratus* (Camaleón).



1. Mecánica

Para la construcción de la estructura del robot se optó por una base de metacrilato de 5 mm. de grosor, ya que este material da una gran resistencia y a la vez permite trabajar sobre él de forma sencilla.

Sobre la base están situados los motores y el sistema de tracción con sus respectivas ruedas y una pequeña torre de aluminio, en la cual se localizan los sensores ópticos



Fig. 1. Motor del robot

2. Tracción

Nuestra elección es el motor paso a paso dada su sencillez de manejo y precisión. Con servos y motores de corriente continua es muy difícil hacer que el microbot haga una trayectoria limpia y precisa. Para ello habría que utilizar mecanismos como decoders para saber cuántas vueltas ha dado el motor.

Elegimos dos motores de 200 p.p.v. (pasos por vuelta) procedentes de impresoras antiguas. Son motores muy precisos y con un par muy grande.

Para conseguir un movimiento fluido, el robot se sustenta en dos ruedas motrices y una rueda loca.

El control de motores se realiza mediante dos integrados de SGS: L297 y L298. Con estos integrados podremos atacar a motores con dos estatores, es decir, motores bipolares de dos fases, o unipolares de 4 fases. La corriente máxima por fase es de 2A. El primero genera

las señales de control para un motor de dos estator; mediante la programación en sus entradas, compatibles TTL, permite la selección del sentido de rotación y del tamaño del paso (paso completo o semipaso). Según sea el caso, el motor avanza o retrocede un paso, o un semipaso en el flanco descendente de la señal aplicada a la entrada. En nuestra placa pondremos un selector mediante el cual se indica si se desea que el motor avance un paso o medio paso cada pulso.

También simplifican el proceso de programación, ya que con solo dos entradas son totalmente gobernados (velocidad y sentido).

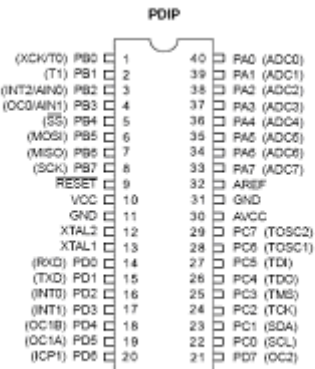


Fig. 5. ATMega32

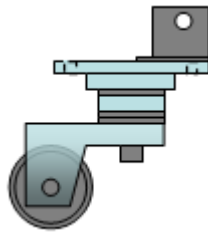


Fig. 2. Rueda Loca

3. Sensores

Usamos dos tipos de sensores:

Para la prueba de laberinto usamos unos sensores Sharp

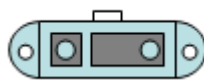


Fig. 3. Sensor Sharp 2Y0A21

Para la prueba de rastreadores usamos los archiconocidos sensores CNY 70

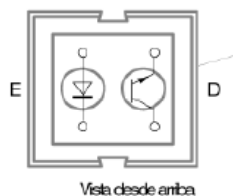


Fig. 4. Sensor Sharp CNY70

4. Microcontrolador

La parte de control reside en el microcontrolador. Para esta tarea hemos utilizado el microcontrolador de Atmel ATMega32.

Éste se encarga de todo el control del robot: sensores, motores, posicionamiento navegación, etc. En su memoria interna están programados todas las rutinas, el plano del laberinto y el algoritmo de salida, que permite que al robot salga en el menor tiempo posible del laberinto. Además, tiene otro programa que se encarga de actuar como rastreador. Seleccionamos el programa que queremos que ejecute mediante un microswitch.

La programación la realizamos en c, con la ayuda de la librería libc para gcc-avr. Esta programación la hacemos bajo Linux.

5. Alimentación

Nuestra elección es una batería de plomo recargable que proporciona 12V y 2.3Ah. Es una batería de la marca Enerbat®. Con esta batería el robot tiene una autonomía variable, desde 4 horas si no se alimentan los motores hasta menos de una hora si se está con ellos encendidos pero sin movimiento. Esta variación se produce porque si los motores están alimentados pero parados son atravesados por intensidades muy grandes, de tal manera que el gasto energético se dispara.