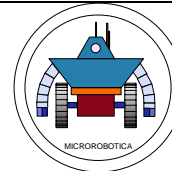
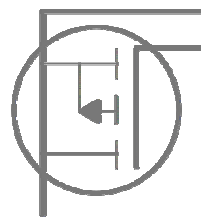




Madrid-bot



Robot FET_0x08



Line Follower Robot Project

Vilches Expósito¹, Sergio; Goitia Hernández², Jon; Castro Castilla³, Gonzalo; García García⁴, Ruy
vilxes91@yahoo.es - jon.goitia91@gmail.com - gonzalo.castro.castilla@gmail.com - rayden.wwe@gmail.com

Universidad Carlos III de Madrid

Centro de referencia: IES Clara Campoamor (Getafe)

Resumen

Este documento describe el desarrollo del robot FET_0x08, diseñado y construido por cuatro estudiantes de primer curso de Grado en Ingeniería Electrónica Industrial y Automática de la Universidad Carlos III de Madrid. En marzo de 2010, el robot participará en las competiciones Cosmobot y Madridbot, siempre en la modalidad de velocistas.

La base del robot es el chasis de un coche radiocontrol, que ha sido modificado y al que se le han añadido sensores para seguir la línea. Toda la parte electrónica ha sido creada desde cero, y se ha desarrollada alrededor del chip Atmega328.

Introducción

FET_0x08 es un robot que aprovecha la mecánica de un coche radiocontrol Iwaver 02 escala 1:28, modificada para poder seguir líneas y ensamblar la electrónica necesaria.

A éste chasis se le cambió el motor de dirección por un microservo, se le añadió una placa con 5 sensores de reflexión IR y una PCB que integra tanto la electrónica de potencia y los reguladores de tensión como la lógica de control (basada en un Atmega328 y programada con Arduino). Para

evitar fallos de suministro al chip se ha optado por utilizar una fuente semi-independiente que se detalla en el apartado correspondiente.

El objetivo de participar en esta prueba ha sido iniciarnos en la robótica de competición y poner en práctica los conocimientos adquiridos hasta ahora.

El equipo

Como se ha dicho, somos cuatro estudiantes que, frustrados por la dura carga teórica de nuestros estudios, decidimos juntarnos e intentar aprender por nuestra cuenta habilidades prácticas. Por ello, adquirimos varios *Arduinos*, (plataforma de computación física *open hardware* de bajo coste), con los que aprendimos las bases de la electrónica y programación de microcontroladores en C.

1. Objetivos de diseño

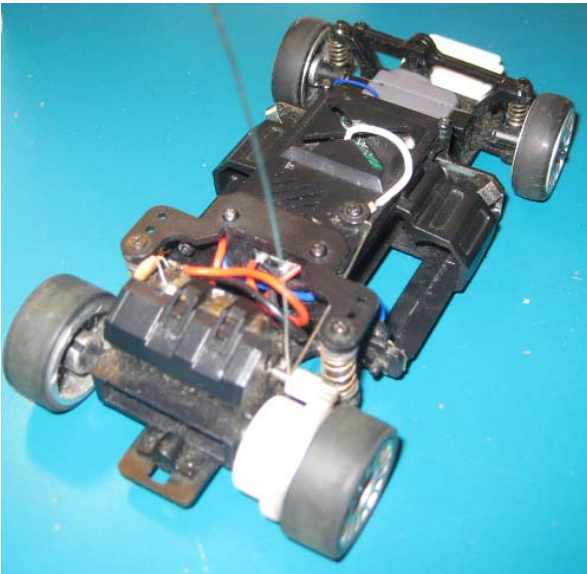
La decisión de participar en la competición de velocistas se tomó debido a que uno de los miembros del grupo tenía a su disposición un coche RC, lo que ofrece ciertas facilidades en cuanto a la mecánica del robot. Por ello, el principal objetivo del diseño ha sido aprovechar y

mejorar al máximo esta mecánica, así reutilizar componentes que tuviésemos en casa para evitar un coste elevado del robot.

2. Estructura mecánica del robot

Como hemos dicho, la mecánica del robot es un chasis de un coche Iwaver 02 escala 1:28 que utiliza propulsión trasera, incluye diferencial, y cuya dirección está controlada mediante un sistema similar al utilizado en automoción. Tras quitar toda la electrónica original, se le modificó el sistema de dirección.

Originalmente, el coche tenía un sistema de dirección activado por un pequeño motor con reductora y un potenciómetro para determinar su posición. Debido a la poca fiabilidad de este mecanismo, la primera mejora fue cambiarlo por un microservo de medidas similares (*Turnigy TG9e*). Además, este cambio hace que el método de control a nivel de programación sea más sencillo.



1. Chasis del coche RC utilizado

3. Sistema sensorial

La tarea de detección de líneas la llevan a cabo 5 sensores de reflexión por infrarrojos QRD1114, colocados de manera simétrica en una PCB diseñada y creada por el equipo. Los sensores envían información analógica al chip para su posterior interpretación.

La placa está diseñada para poder cambiar la configuración de los sensores. Hay 3 sensores fijos (en el centro) para detectar la línea en los trayectos rectos, y otros dos sensores laterales con posiciones variables, gracias a un conjunto de 8 zócalos donde pueden colocarse.



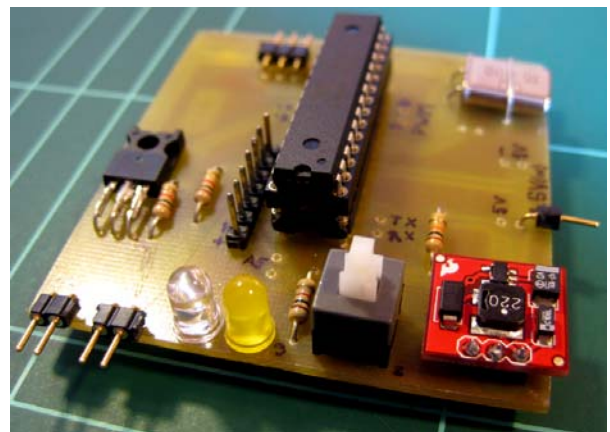
2. Placa de sensores

4. Sistema de tracción

El robot emplea un motor DC de 6V. Se puede hallar más información sobre el motor en www.iwaver.com/iwaverparts-02-ogi-1.html

La velocidad del motor se controlará a través de *Pulse Width Modulation* (PWM), que consiste en variar el suministro de corriente al motor muchas veces por segundo, consiguiendo así un valor intermedio que puede variar entre la velocidad mínima (parado) y la velocidad máxima.

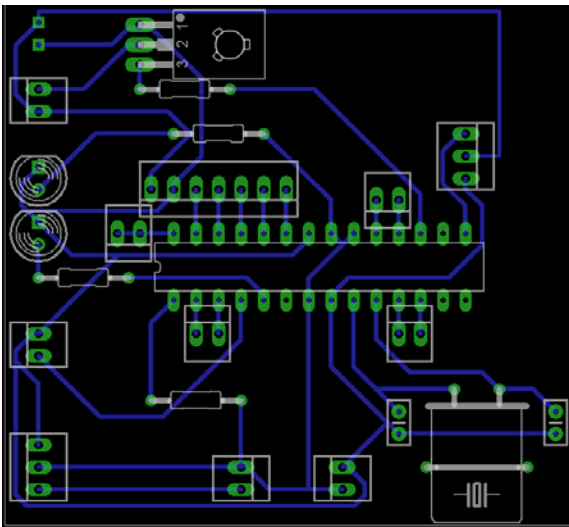
5. Electrónica



3. Placa principal con microcontrolador

La electrónica de este robot ha sido diseñada y creada desde cero por los miembros del grupo. Como ya se ha dicho, el procesador es un Atmega328 programado con Arduino. La información analógica enviada por los sensores es interpretada por el chip, que a su vez envía los impulsos necesarios al motor y al servo para

controlar la velocidad y la dirección del robot. La placa está diseñada para hacer posible futuras modificaciones y mejoras. Por ejemplo, se han instalado dos LEDs, uno para indicar que el circuito está conectado a la batería, y otro que es programable, para poder cambiar su significado. También se han añadido extensiones de algunos pines del chip que no están en uso, para facilitar la añadidura de otros componentes si fuese necesario. Ambas PCBs (la principal y la de los sensores) fueron diseñadas con el software gratuito Eagle CAD.



4. Diseño de la placa principal con Eagle CAD

6. Sistema de Alimentación

El robot lleva cuatro pilas AAA (1.5 V cada una) conectadas en serie para formar una batería de alimentación de 6V. El motor, el servo y los sensores están conectados directamente a esta fuente. Sin embargo, como el procesador opera a 5V, no se puede conectar directamente. Se ha decidido utilizar un transformador DC/DC, que convierte 2.5 V (conectando dos pilas) en 5V. De esta manera, hay dos pilas que, además de contribuir a la batería principal, también están alimentando el procesador y los dos LEDs.

7. Programación

El robot se sirve de los cálculos de un procesador avr, idéntico a los empleados en Arduino. Para la programación del mismo hemos empleado el software proporcionado por la comunidad de Arduino, que consiste en un IDE de programación. El procesador se programa en C,

sin embargo, sólo se tienen que definir dos funciones para poder hacer funcionar el programa: setup que corre una vez al inicio (se calibran los sensores y se inicializan las configuraciones) y loop, que se ejecuta repetidamente.

8. La organización del trabajo del equipo

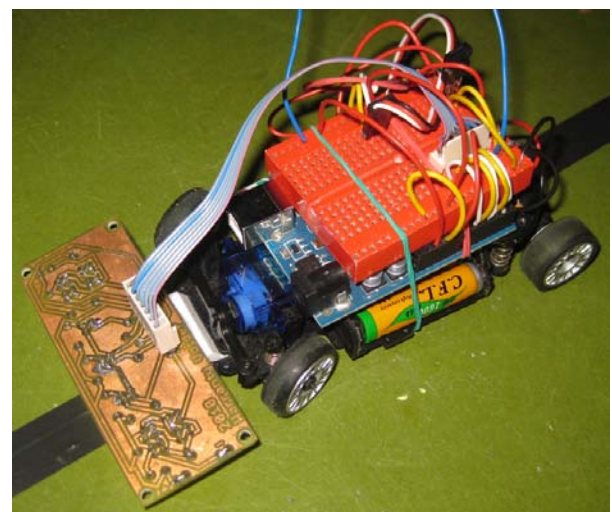
El trabajo en equipo se ha visto limitado por la distancia entre los diferentes miembros del grupo.

Por ello, la coordinación de trabajo se llevó a cabo por medio de correo electrónico y una Wiki sobre el robot, a través de la cual recopilábamos información y opiniones.

En las últimas fases del desarrollo, optamos por hacer la mayor parte del trabajo restante en grupo, para así poder colaborar todos en el diseño creativo y búsqueda de soluciones a los problemas.

Las fases que se han seguido han sido son las siguientes:

- Ingeniería inversa sobre el coche RC para averiguar los parámetros de funcionamiento.
- Búsqueda de materiales y recursos para adaptarlo a nuestras necesidades.
- Creación de un prototipo para poder probar electrónica/mecánica y probar con diferentes ajustes.
- Creación de la electrónica final.
- Desarrollo del código.
- Mejora del código.



5. Fase de prototipado

9. Problemas encontrados y posibles mejoras

La mayor dificultad con la que hemos tenido que tratar es con el poco agarre de los neumáticos del coche RC con el suelo. Hasta ahora, no hemos encontrado ninguna solución a este problema, ni parece que haya ninguna, ya que estos coches están diseñados para correr sobre pistas de lija, y no superficies lisas.

Por otra parte, el sistema de alimentación que el coche trae de serie (4 pilas AAA) resulta ser poco efectivo. Una alimentación basada en baterías LiPo habría tenido mejor resultado.

10. Conclusiones

Este trabajo nos ha servido, ante todo, para ver que las cosas no son tan fáciles como parece. Algo tan simple como seguir una línea puede llevar muchos quebraderos de cabeza, algunos bastante previsibles (el algoritmo de control), pero otros inesperados (bajadas de tensión, ruido debido al motor, fallos eléctricos y mecánicos, etc).

Lo bueno: que cada uno de estos obstáculos te enseña algo diferente, y te ayuda a no volver a cometer los mismos errores.

Por otra parte hay que mencionar que, aún habiendo momentos frustrantes, el desarrollo del robot ha sido bastante divertido, al mismo tiempo que desafiante.