

# Competencia de Robótica R2-D2 2014

Categoría: **Mini-Sumo**

Nombre del Robot: **El Loco**

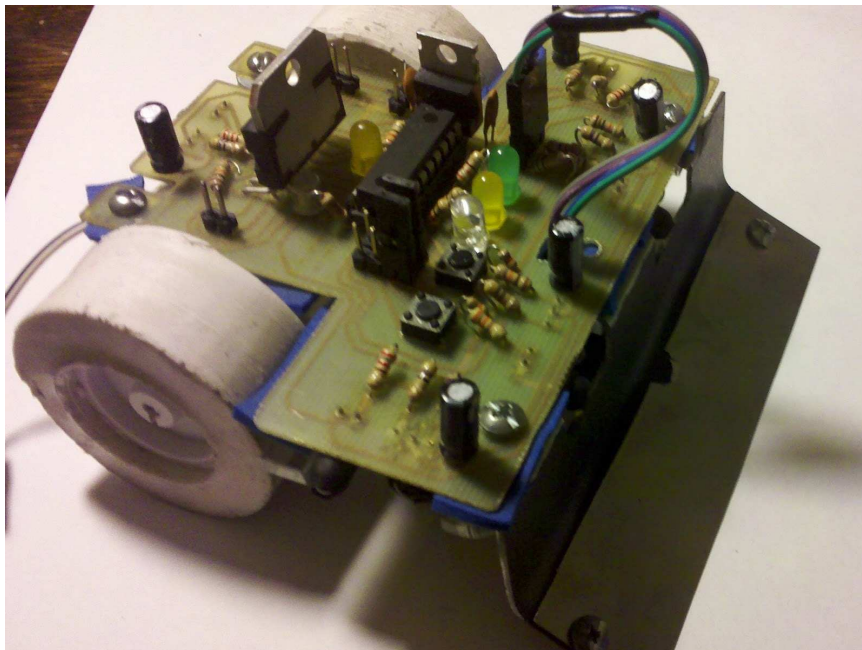
Institución: **Independiente**

Participantes:

Ezequiel Canay

Martin Di Lullo

Teo Gentile



## INTRODUCCIÓN:

Todo el robot esta diseñado y armado en base a 2 objetivos:

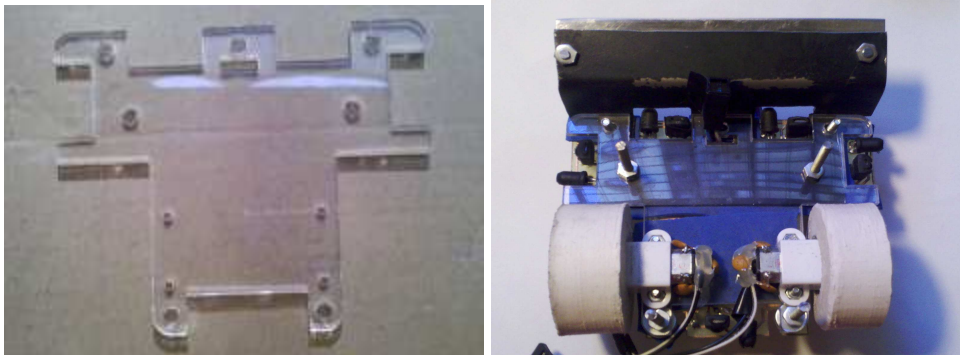
- **Minimizar el espacio:** Para generar una estructura con una baja altura, para tener un centro de masa bajo y dificultar la detección de los otros robot.
- **Disminuir el error humano en el armado:** creemos que en nuestro equipo una de nuestras principales debilidades es nuestra capacidad manual, sumado a que en la categoría Mini-Sumo se tiene un muy reducido espacio, el armado preciso es muy importante.

## MECÁNICA:

### Chasis:

Con la idea de “**minimizar el espacio**”<sup>1</sup>. El robot está armado usando como base una sola pieza. A la cual se le adhiere todas las otras partes mediante tornillos y tuercas. Creemos que esto optimiza el espacio eficientemente, disminuyendo el que ocupa las piezas de apoyo.

En base a la idea de “**disminuir el error humano en el armado**”, la base esta hecha de acrílico. Fue diseñada por computadora, en adobe Illustrator CS6 y mandada a cortar con láser. Esto descarta el error humano en las medidas, que normalmente genera un descentrado de la tracción, que perjudica el funcionamiento mecánico del robot.

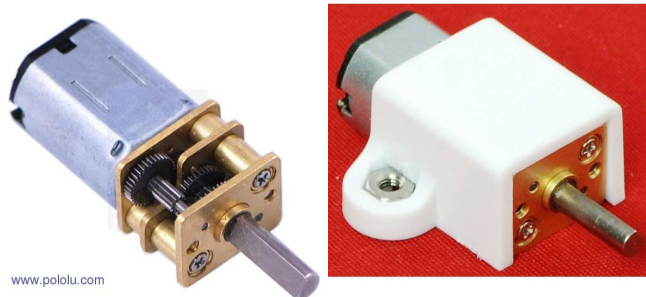


### Motores:

Utilizamos dos Micro Metal Gearmotor HP, de reducción 150:1, comprados en la página [www.pololu.com](http://www.pololu.com). Gracias a su buen rendimiento y su pequeño tamaño **optimizan el espacio utilizado**. Además compramos en la misma página unos soportes que vienen especialmente diseñados para esos motores, **disminuyendo el error humano en el armado**, generando un muy buen agarre optimizando el espacio.

---

<sup>1</sup> Verde: Ventaja.



### Ruedas:

Inicialmente usamos las ruedas que vienen especialmente diseñadas para estos motores, que poseen un encastre perfecto a presión y que **disminuye el error humano en el armado**. Pero le vimos el defecto de que la goma con la que vienen tiene una muy poca adherencia. Por lo cual diseñamos un molde por computadora y lo imprimimos en una impresora 3D, de esta manera le pusimos a la llanta de la rueda una cubierta de silicona, conservando el encastre y aumentando la adherencia.



### Baterías:

Usamos 2 baterías de lipo, de celular. gracias a su poco tamaño y a su gran capacidad. son lo más eficiente con respecto espacio rendimiento que encontramos. Siendo la mejor opción para **minimizar el espacio** pero lamentablemente **aumenta el error humano<sup>2</sup>**, porque genera muchos problemas la frágil conexión necesaria para poner las 2 baterías en series y que es necesario cargarlas usando un celular.

## **ELECTRÓNICA:**

### Controlador:

Usamos un pic 16f628a para controlar el comportamiento del robot y los sensores. Elegimos este pic por 2 motivos,

---

<sup>2</sup> Rojo: Desventaja

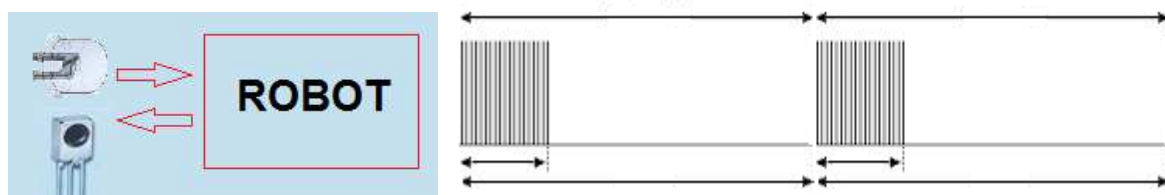
- Por su tamaño, que disminuye las dimensiones de la placa, ayudando a **minimizar el espacio**.
- Principalmente porque ya veníamos trabajando con él, haciendo mucho más fácil el trabajo, ya que estamos muy familiarizados con su funcionamiento.

La principal desventaja que le encontramos es que por su **limitada cantidad de entradas y salidas**, solo llegamos a controlar las cosas “indispensables” y nos deja con una mala comunicación humano-robot. Tenemos pensado cambiar de controlador para poder agregarle una comunicación mediante un puerto serial con una pc.

### Sensores:

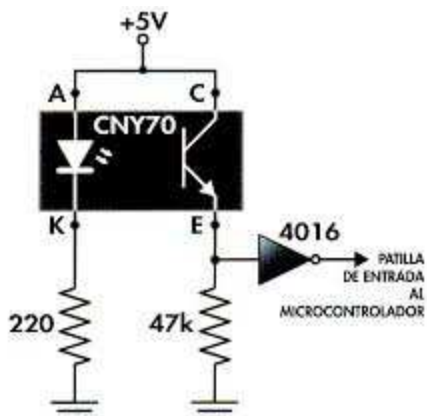
#### Proximidad:

Usamos 5 sensores para detectar la proximidad de los otros robots. Dos para adelante, uno para cada lado y uno para atrás. Están formados por un receptor infrarrojo de control remoto (IRM-8602S). Trabajando con un led emisor infrarrojo, el cual emite pulsos de señales cuadradas de 38 KHz, mediante el PWM del PIC, las cuales rebotan contra los otros robots y llegan al receptor. Estos sensores son más **simples y económicos** en comparación con los sensores de ultrasonido. Pero tiene mucha **variabilidad en la distancia de detección** con respecto al color del objetivo.



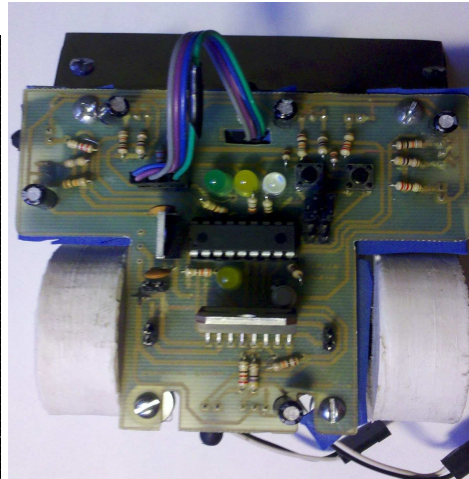
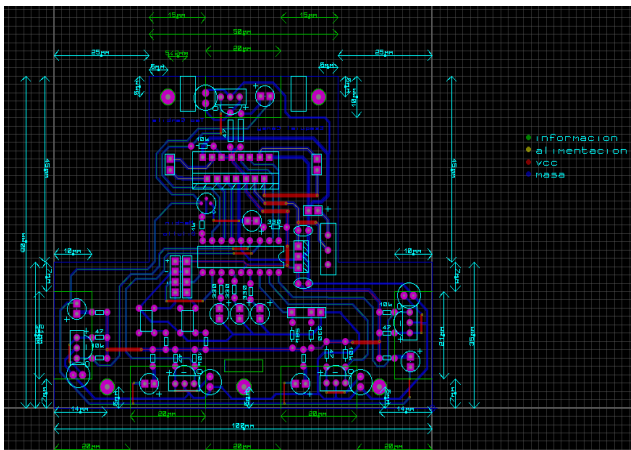
#### Línea:

Usamos un CNY70, que viene con un led infrarrojo y fototransistor. La luz rebota y ilumina el fototransistor, el cual genera una señal analógica con respecto al nivel de la luz rebotada (blanco rebota más que negro). Nosotros mediante una resistencia de pull-up la utilizamos como una señal digital.



## Placa:

Cuando diseñamos la placa intentamos **minimizar el espacio**, para lo cual decidimos poner en la misma placa todos los componentes, no hacer los sensores por separado en otra placa y conectarlos mediante cables. Esto optimiza el espacio, ganando el que ocuparía las placas de los sensores y el agarre de ellas. También **disminuye el error humano en el armado**, sacando la necesidad del agarre y los cables. Además hicimos que los sensores salgan por debajo de la placa. para que estén mas abajo y puedan detectar a robot bajos.

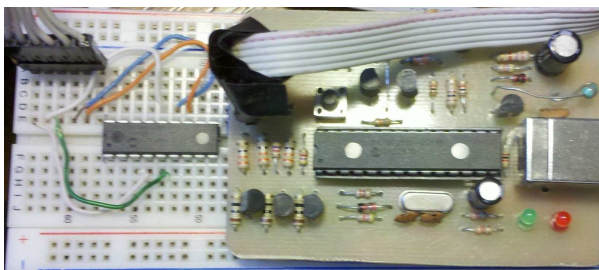


## **PROGRAMACIÓN:**

### Herramientas:

Usamos el MPLAB IDE v8.92 como entorno de desarrollo, porque es el que estamos acostumbrados a usar para programar PIC. No conocemos otros como para comparar el funcionamiento, pero nunca tuvimos ningún problema.

Utilizamos como programador un clon del PICkit 2, hecho por nosotros. Como tenemos en uso las patas del pic que son para programar, **tenemos que sacarlo de la placa**, lo colocamos en un protoboard y lo conectamos mediante cables al clon de PICkit 2.



### Programa:

El Loco

Programamos en el lenguaje C, utilizando el método de máquinas de estados. El programa esta formado por 3 maquinas:

1. La que controla los sensores de proximidad, que se encarga de las emisiones de los led, y de decidir con respecto a cuántos pulsos volvió a recibir, si detecto algún robot.
2. La parte de testeo de funcionamiento, antes de que robot empiece a “pelear” testeamos el funcionamiento de los sensores, **como no tenemos un led por cada sensor**, tenemos que hacer la prueba de a partes, primero los 2 de adelante y el del piso y despues lo de los costados y el de atrás.
3. La máquina que controla el comportamiento del robot en la pelea. usando la información de que detecta cada sensor.

## CONCLUSIÓN:

Estamos muy **conformes con el desarrollo** del robot, gracias a que es el primer robot que hacemos fuera del ámbito escolar, pudimos organizarnos y solucionar muy bien los problemas imprevistos. El mayor problema que le encontramos al diseño del robot, fue **la limitada cantidad de entradas y salidas que tiene el controlador que usamos**. Porque generó varios problemas, como la necesidad de hacer el testeo en dos etapas porque no tenemos suficientes led, la utilización de solo un sensor de piso y nos impide implementar la mayoría de las mejoras que se nos fueron ocurriendo a lo largo del proyecto, como conectar un puerto serial a la placa y no tener que sacar el controlador para programarlo.

## Costo:

Gracias a que encargamos los motores, los agarres y las ruedas el año pasado en el “Black Friday” pudimos ahorrar mucho, gastando \$350 en eso. La placa salió alrededor de los \$200 y la base de acrilico \$40. El robot salió en su totalidad aproximadamente \$600.