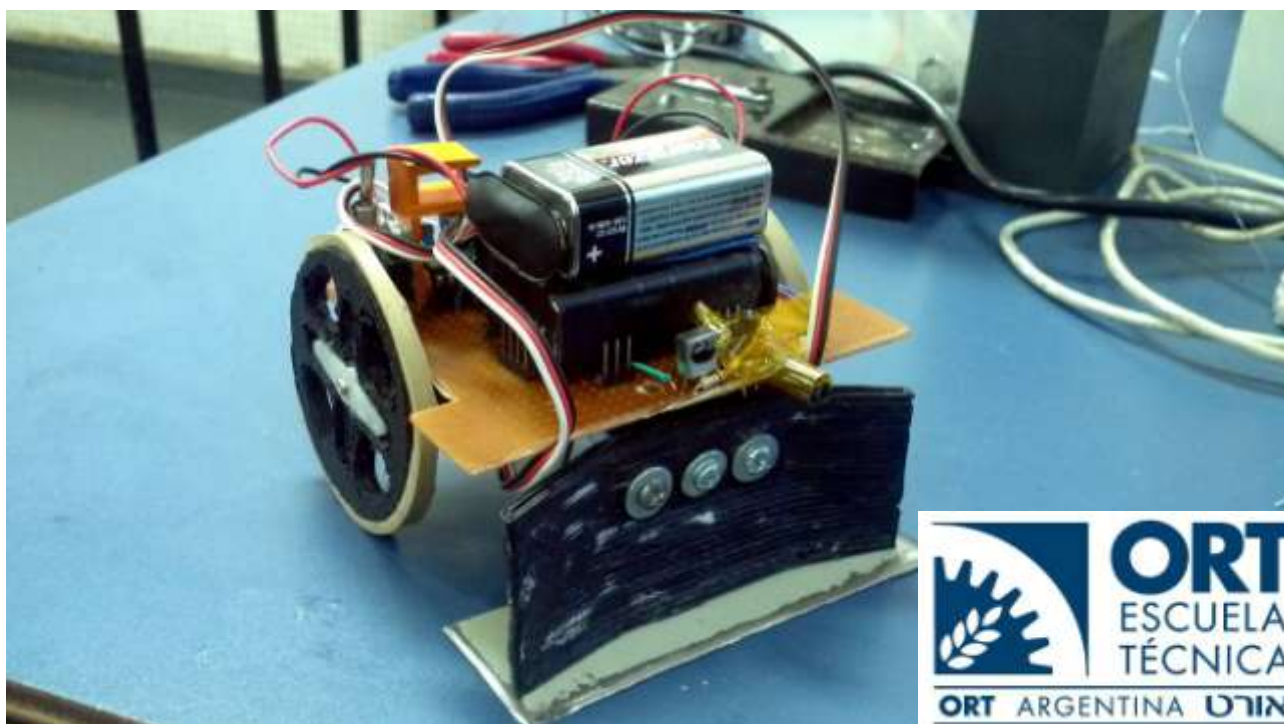
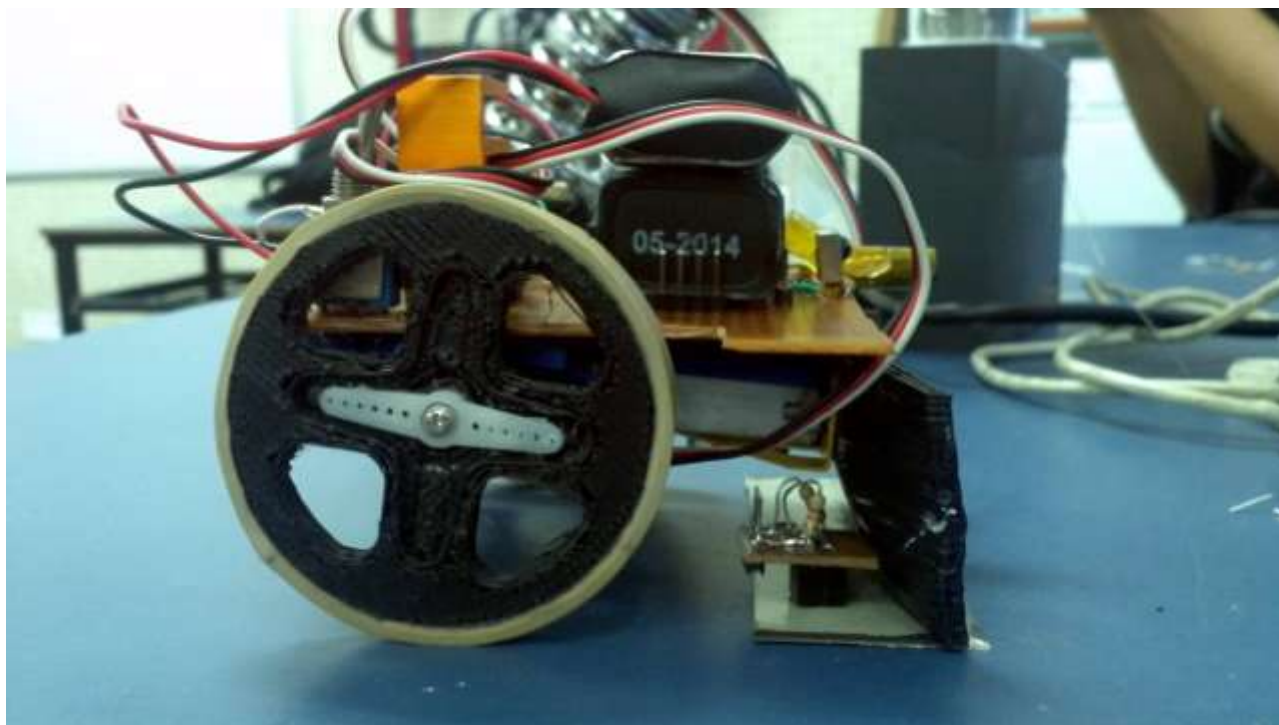


# Optimus 3D

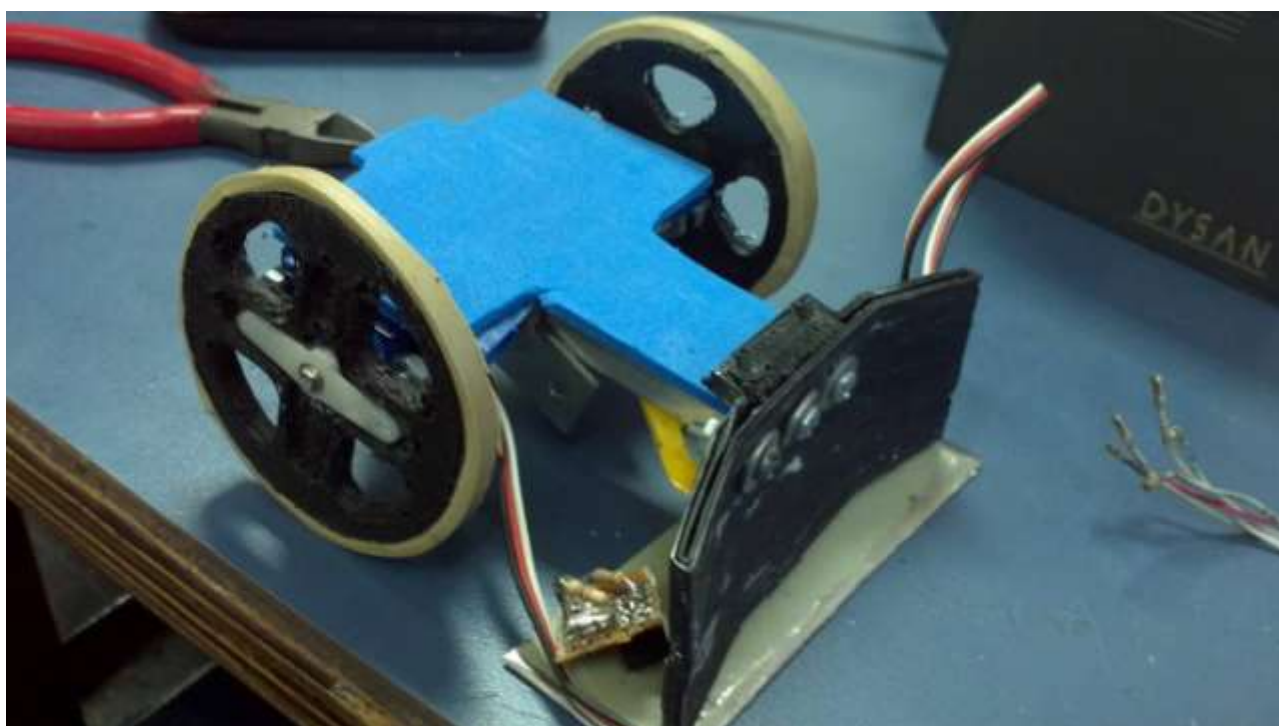
Escuela Técnica ORT, sede Almagro  
Gino Tubaro



## Mecánica



Vista lateral.



Mecánica.

## Características

- Dimensiones: 100 x 100mm. Altura 60mm
- Ruedas de 50mm de diámetro
- Estructura: plaqueta universal para prototipos electrónicos + impresiones 3D para rampas y ruedas sujeto a perfiles de aluminio buscando simplicidad y reducción de peso(100 x 100 mm)
- Tracción diferencial (dos servos trucados en la parte posterior)
- Rampa como punto de apoyo.
- Peso: 450 gramos

## Electrónica

- Controlador PIC 16F876A(circuito esquemático en anexo)
- Controlador de motor L293D, 1A de alimentación y 5v mas 5v lógicos para operar.
- Sensores infra-rojos 3 CNY70 para detectar piso y un modulo receptor infra-rojo de 38kHz (circuito esquemático en anexo)
- Reguladores independientes (7805)

## Funcionamiento básico

Básicamente el funcionamiento tiene 2 etapas durante:

### **Testeo:**

El robot en su principio inicia el programa testeando los motores y los sensores. Es un bucle infinito que se interrumpe cuando se inicia la competencia. El led rojo indica cuando uno de los cuatro sensores esta activado.

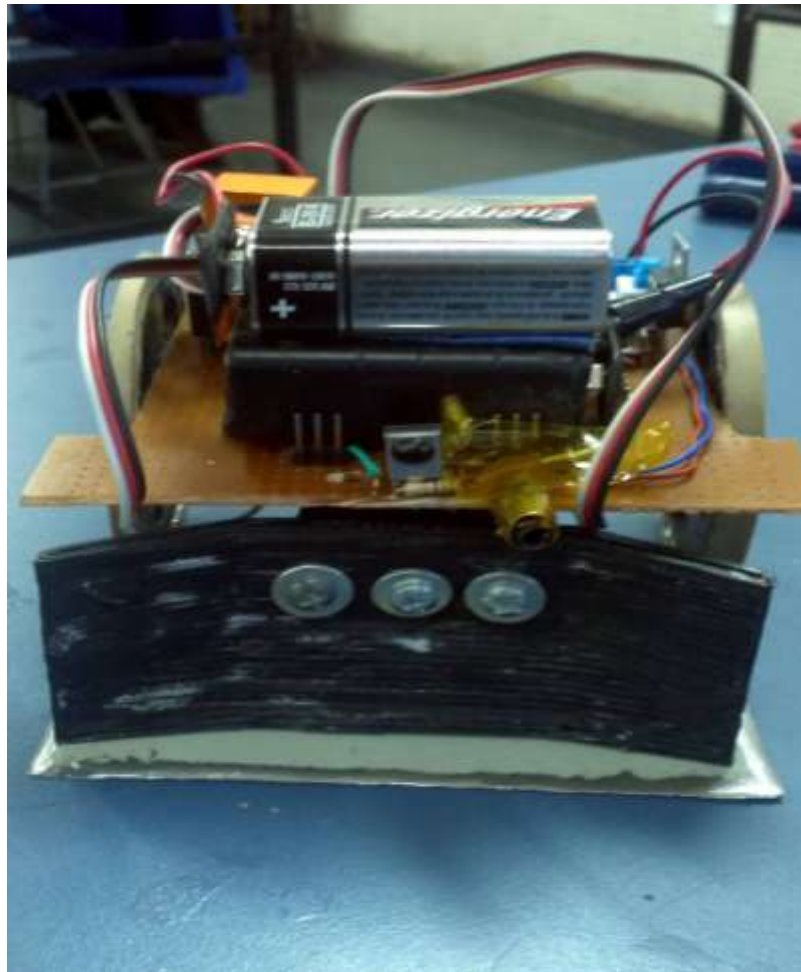
### **Combate:**

Durante el combate el robot va a buscar a su oponente, con una señal de PWM el led infra-rojo emite en 38kHz que, con un alcance de 25 cm, se utiliza para localizar al robot enemigo. Cuando el haz de luz refleja sobre la superficie del contrincante, este rebota, siendo captado por el modulo de

infra-rojo. Este modulo comunica la ubicación del robot contrincante para que Optimus 3D ataque.

Para que Optimus 3D no salga del tatami utiliza 3 sensores de infrarrojo, dos ubicados en su frente para ser más preciso al momento de rebotar en la línea curvada y uno en su parte posterior para detectar en un caso extremo de estar siendo empujado al borde del tatami.

En el combate, además, el L293D se encarga de controlar los motores, tiene su propia fuente regulada de 5V continuos con un LM7805 y un disipador para soportar la temperatura y tener una velocidad constante.





## Componentes y precios

**Costo total: \$100**

**Ruedas: Impresas en 3D (costo proporcional al plástico usado 6,7\$/u)**



<http://www.thingiverse.com/thing:21486>

El diseño de la misma fue basado descargado de [Thingiverse](http://www.thingiverse.com/thing:21486) modificado con OpenSCAD e impreso con una RepRap Prusa Mendel ensamblada en casa. Con una duración de impresión de una hora cada rueda. Tiene en su interior un agujero adaptable al servo trucado. Además, cada rueda pesa menos de 5 gramos y son altamente resistentes a golpes.

Diseño en OpenSCAD



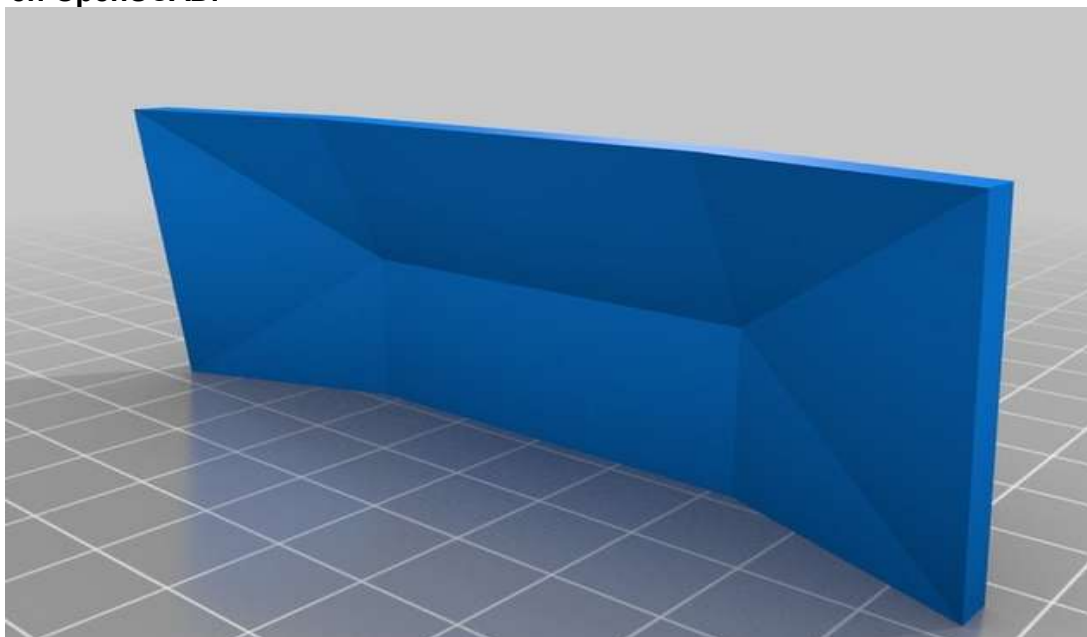
**Rampa: Impresa en 3D (costo proporcional al plástico usado 8\$/u)**



<http://www.thingiverse.com/thing:29827>

El diseño de la misma fue basado descargado de Thingiverse modificado con OpenSCAD e impreso con una RepRap Prusa Mendel ensamblada en casa. Con una duración de impresión de media hora. Fue mecanizada para conectarse a la estructura de aluminio. Además, pesa casi 10 gramos y es altamente resistente a golpes.

**Diseño en OpenSCAD.**

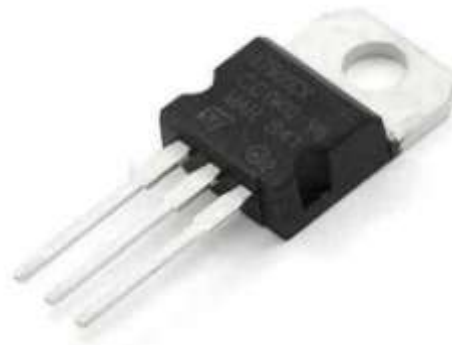


**PIC16f876A (45\$)**

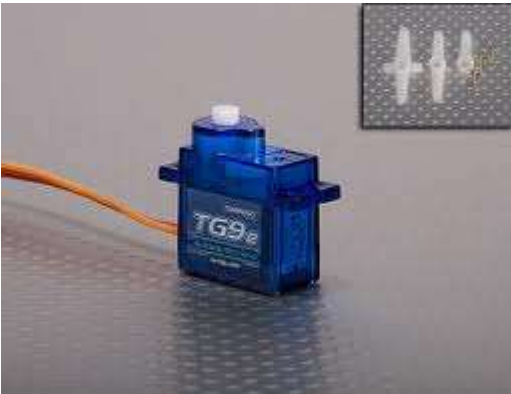


<http://articulo.mercadolibre.com.ar/MLA-458480518-pic16f876a-pic-micro-microcontrolador-pic16f- JM>

**LM7805 (7\$)**



**Servo motor trucado(\$40)**



**L293D(\$40)**



### Bateria 9v (\$12)

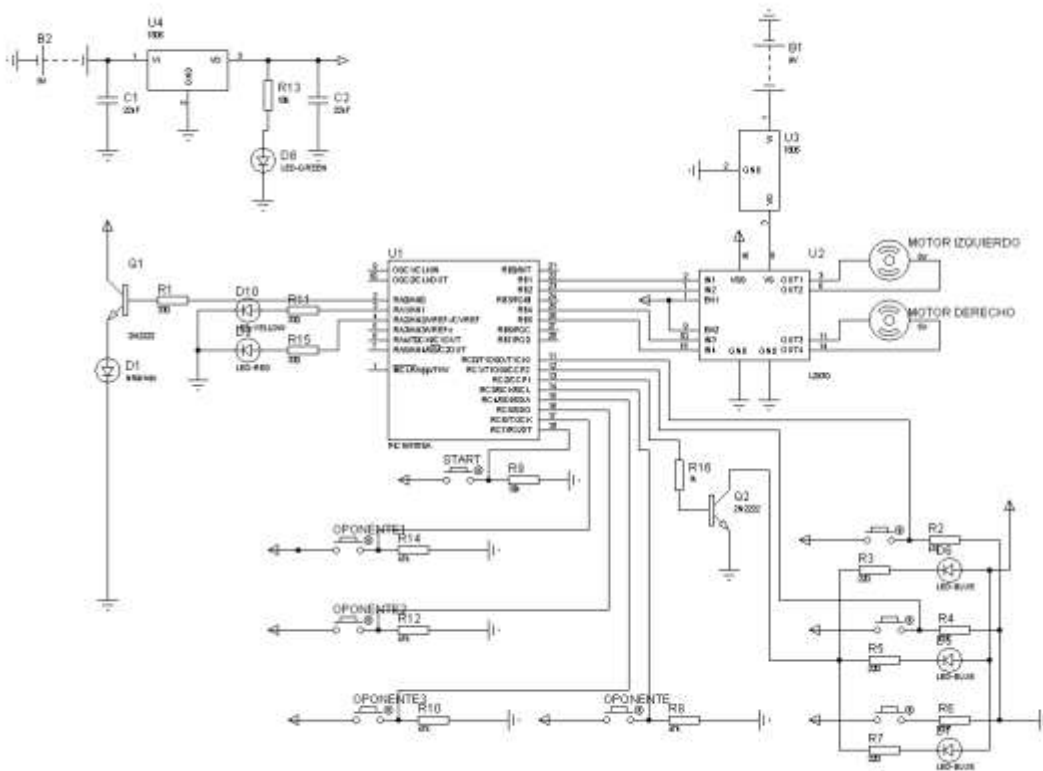


### Plaqueta experimental (10x10cm) (\$30)



[http://articulo.mercadolibre.com.ar/MLA-460698723-plaquetas-experimentales-pertinax-10x10cm-high-tec-electroni-\\_JM](http://articulo.mercadolibre.com.ar/MLA-460698723-plaquetas-experimentales-pertinax-10x10cm-high-tec-electroni-_JM)

## Circuito Esquemático





## Código fuente

```
//PROVISORIO

#include <optimus.h>
#define LED PIN_A0
#define LED2 PIN_A1
#define DELAY 1000
#define ir1 PIN_B
#define int_RB
void RB_isr(void)
{
output_high(LED2);
}

void main()
{
setup_timer_2(T2_DIV_BY_16,255,1); //819 us overflow, 819 us interrupt
setup_ccp1(CCP_PWM);
set_pwm1_duty((int16)510);
setup_comparator(NC_NC_NC_NC);
enable_interrupts(INT_RB);
enable_interrupts(GLOBAL);

//Example blinking LED program
while(true)
{
output_low(LED);
delay_ms(DELAY);
output_high(LED);
delay_ms(DELAY);
}
}
```