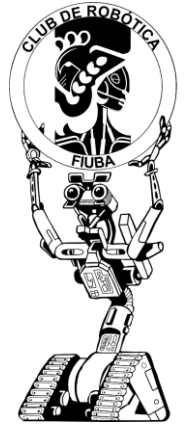


Competencia de robótica

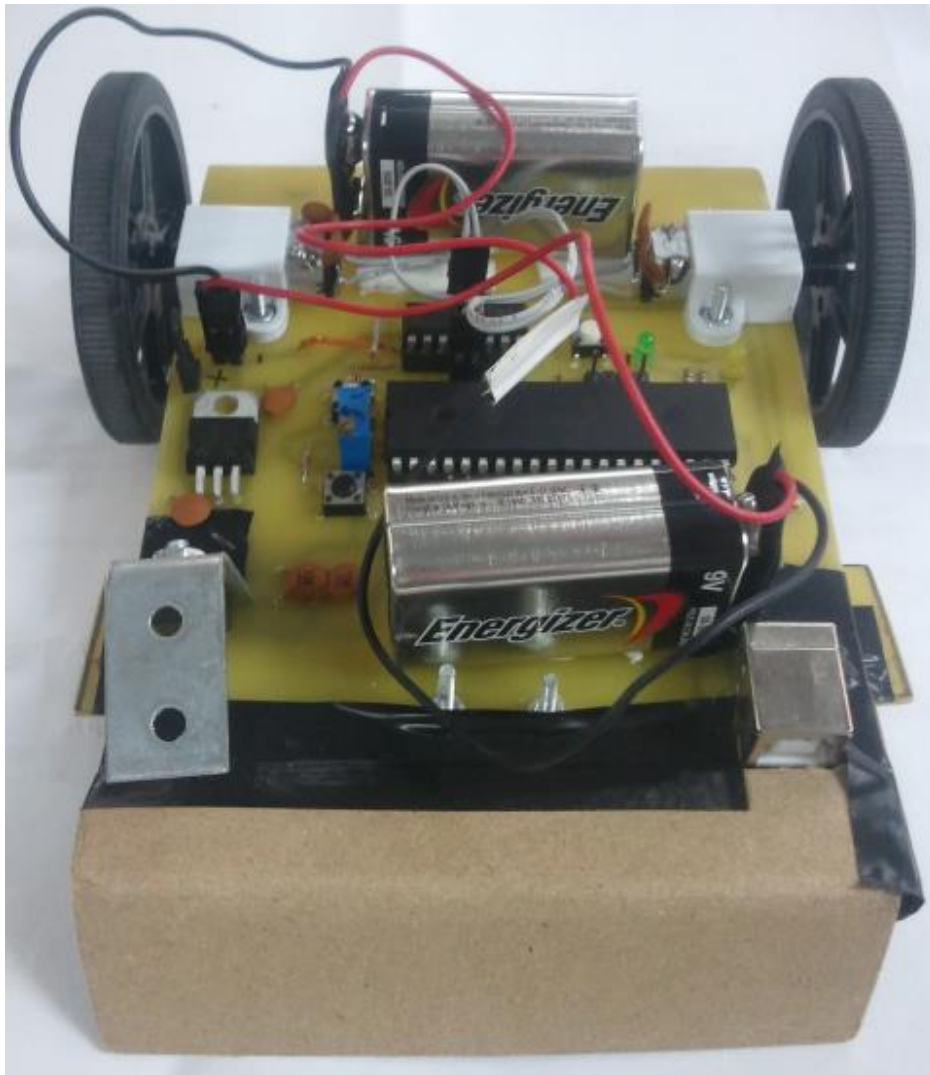
JOHNNY 5

~2015~



Categoría: **Velocista**

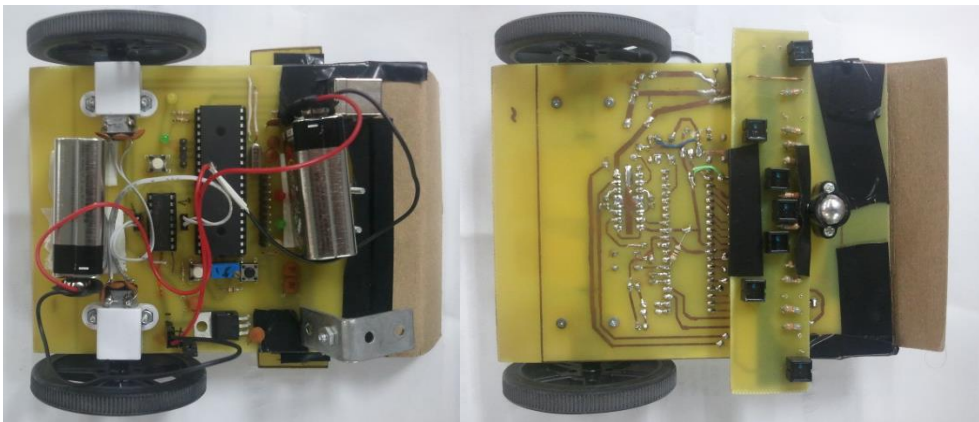
Nombre robot: **Bigote**



Institución: **Escuela ORT, sede Almagro**

Mariano Fouiller (mfouiller@ort.edu.ar)

Mecánica



Vista superior

Vista Inferior

- Dimensiones: 120 x 150mm. Altura 60mm
- 2 Ruedas de 60mm de diámetro
- Estructura: plaqueta de circuito impreso FR4 (10x13cm)
- Tracción diferencial (dos motores en la parte posterior)
- Punto de apoyo, tipo ball-caster
- Peso: 246 g

Electrónica

- Micro Controlador PIC18F4550
- Driver de motores dc L293D
- Regulador lineal 5v (7805)
- Regulador lineal 12v (7812)
- 3 Sensores reflectivos CNY70
- 2 Baterias 9v

Funcionamiento básico

Básicamente el funcionamiento tiene 4 etapas:

Calibración de los sensores:

Se utilizan 3 sensores, cada sensor responde de manera parecida pero no idéntica al color detectado (emiten luz infrarroja y miden la intensidad de la luz recibida). Al presionar el pulsador, el robot barre 180° con sus motores, y durante ese movimiento captura muchas muestras de cada sensor, obteniendo al final un valor máximo y otro mínimo para cada uno individualmente. Estos valores son almacenados en RAM y luego utilizados.

Luego, se presiona una vez más el pulsador y el programa entra en un loop continuo ejecutando 2 funciones principales:

Adquisición, ajuste y procesamiento de la posición:

Se leen los sensores, utilizando los máximos y mínimos obtenidos en el proceso de calibración, se ajusta la lectura de cada sensor, con una ecuación lineal a un valor entre 0 y 1000 (0 totalmente negro – 1000 totalmente blanco). Con esos 3 valores entre 0 y 1000, se realiza una cuenta, ponderando la posición de los sensores y su valor, para obtener finalmente un número entero entre 0 y 2000 que indica el lugar donde se encuentra el robot respecto de la línea, siendo 0, 1000, 2000 los valores para los cuales el sensor 0, 1, 2, está justo arriba de la línea blanca respectivamente.

Aplicación de ecuación de corrección:

Al ser el sensor 1 el “centro” del robot, nuestro objetivo es mantener el valor de la lectura en 1000. Para esto se resta el valor 1000 al valor calculado (Línea), quedando un valor entre -1000 y 1000, siendo los extremos derecho e izquierdo, y el centro “0”.

Este valor (entre -1000 y 1000) indica cuan lejos estamos del valor ideal (0), por lo que lo multiplicamos por un factor que indica que debemos hacer con los motores en función de cuan alejados estamos del centro. A esto se lo llama constante proporcional. Además, se calcula la diferencia entre los dos últimos valores de la línea, y se obtiene un término derivativo, que nos permite tener una idea de la variación entre una lectura y la siguiente. Este factor, multiplicado por la constante derivativa, se suma también a la corrección.

Modificación de velocidad de los motores:

El valor obtenido se considera la diferencia de velocidad entre el motor derecho e izquierdo. Un motor siempre irá al máximo y el otro más lento (según lo calculado por la ecuación)

Componentes y precios

Costo total:

Microcontrolador PIC 18F4550 (\$85)



<http://articulo.mercadolibre.com.ar/MLA-463177311-pic-18f4550-pic18f4550-pic18f4550-ip-usb-flash-microchip-JM>

Driver motores dc L293D (\$70)



<http://articulo.mercadolibre.com.ar/MLA-460868479-l293d-l293-puente-medio-h-motores-robotica-hobby-JM>

Micro Motores 50:1 (\$170)



<http://www.pololu.com/catalog/product/1098>

Soporte para motores (\$26)



<http://www.pololu.com/catalog/product/1089>

Ruedas (\$42)



<http://www.pololu.com/catalog/product/1420>

Punto de apoyo – rueda loca (\$16)

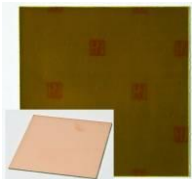


<http://www.pololu.com/catalog/product/951>

(2) Bateria 9v (\$26)



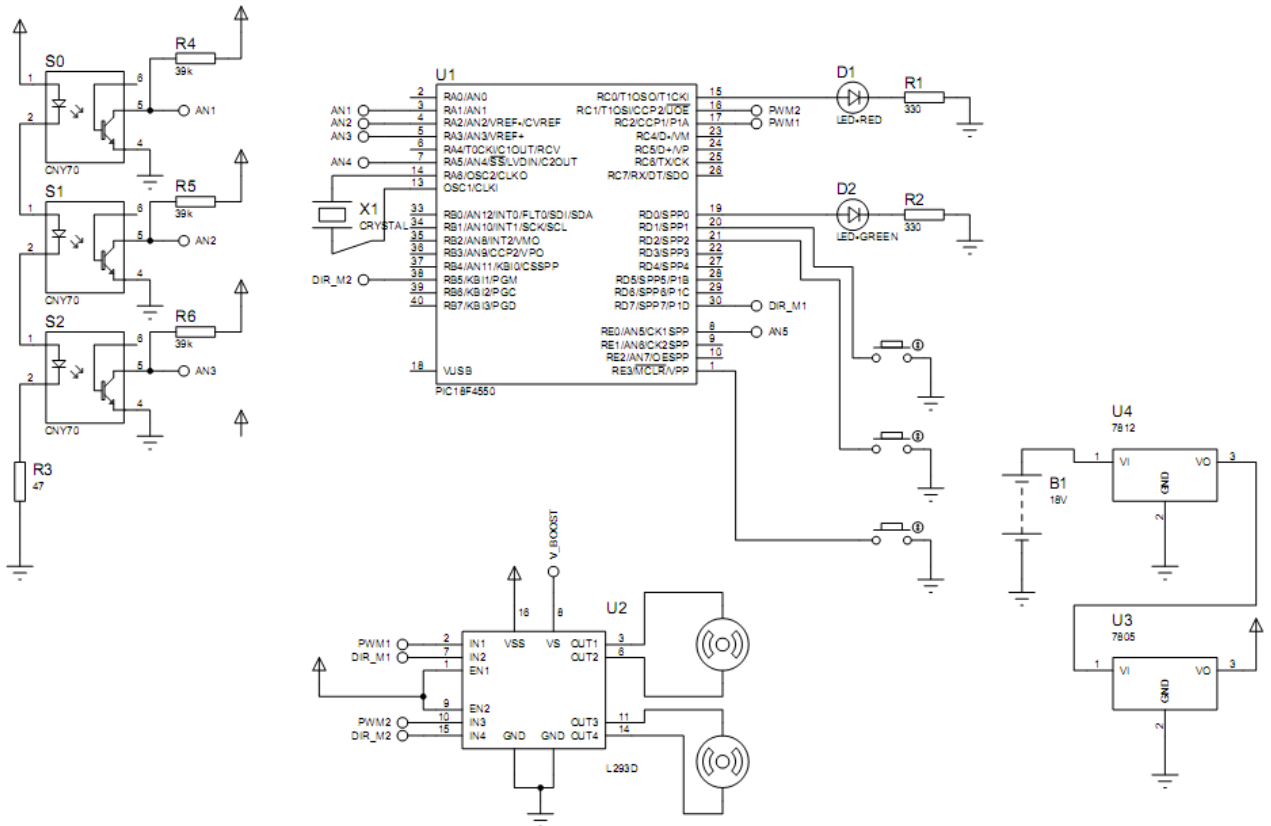
Placa circuito impreso FR4 (\$20)



http://articulo.mercadolibre.com.ar/MLA-463315256-plaquetas-virgenes-epoxi-10x10-cm-de-fibra-de-vidrio-oferta- _JM

Circuitos Esquemáticos

Placa Principal



Código fuente

```
#include <18F4550.h>
#device adc=10

#FUSES NOWDT           //No Watch Dog Timer
#FUSES WDT128         //Watch Dog Timer uses 1:128 Postscale
#FUSES PLL1           //No PLL PreScaler
#FUSES CPUDIV1        //No System Clock Postscaler
#FUSES HS              //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOFCMEN        //Fail-safe clock monitor disabled
#FUSES NOIESO         //Internal External Switch Over mode disabled
#FUSES PUT            //Power Up Timer
#FUSES NOBROWNOUT     //No brownout reset
#FUSES BORV45         //Brownout reset at 4.5V
#FUSES NOVREGEN       //USB voltage regulator disabled
#FUSES NOPBADEN       //PORTB pins are configured as digital I/O on RESET
#FUSES NOLVP          //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOXINST        //Extended set extension and Indexed Addressing mode disabled (Legacy mode)
#use delay(clock=2000000)

##build(reset=0x1000)
##org 0x0000,0x0FFF {}

#define LED_ROJO PIN_D0
#define LED_VERDE PIN_D1
#define LED_AMARILLO PIN_D2
#define LED_VERDE2 PIN_D3
#define IN_M1 PIN_B0
#define IN_M2 PIN_D7
#define BOTON1 PIN_B4
#define BOTON2 PIN_D4

//-----
##define VEL 950
//-----

#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)

void motores(long izq,long der);
void calibracion(void);
void calibrar(void);
void inicializar(void);
void test(void);
void Leer_Linea(void);
```

```
signed long power_difference,VEL;  
long lectura[8],calibrado[8][2],izq,der;  
signed long linea;  
signed long prop,ult_prop=0;
```

```
void main()  
{  
long var;  
long long aux;  
int16 cont,pote,contrecta,contrecta2;
```

```
//-----  
setup_adc_ports(AN0_TO_AN7|VSS_VDD);  
setup_adc(ADC_CLOCK_internal);  
setup_psp(PSP_DISABLED);  
setup_spi(SPI_SS_DISABLED);  
setup_wdt(WDT_OFF);  
setup_timer_0(RTCC_INTERNAL);  
setup_timer_1(T1_DISABLED);  
setup_timer_2(T2_DIV_BY_1,249,1);  
setup_timer_3(T3_DISABLED|T3_DIV_BY_1);  
setup_ccp1(CCP_PWM);  
setup_ccp2(CCP_PWM);  
setup_comparator(NC_NC_NC_NC);  
setup_vref(FALSE);
```

```
set_pwm1_duty(0);  
set_pwm2_duty(0);  
//-----
```

```
test();
```

```
/* while(1)  
{  
set_adc_channel(0);  
delay_us(20);  
pote=read_adc();  
delay_us(30);  
if (pote>1000) pote=1000;  
set_pwm2_duty(pote);  
//delay_ms(100);  
}*/
```

```
while (input(BOTON1)==1){  
inicializar();
```



```

calibracion();

output_high(LED_VERDE2);

while (input(BOTON2)==1)

{
Leer_Linea();
prop=linea-1000;
if (prop<0) {output_high(led_rojo); output_low(led_verde);}
else {output_low(led_rojo); output_high(led_verde);}

}
test();
while (input(BOTON2)==0){

var=70;
aux=(long long)var*1000;

motores(300,300);
delay_ms(50);

cont=0;
VEL=500;
Linea=1000;
while(TRUE)
{

//if (cont<100) cont++;
//else {output_toggle(led_amarillo);cont=0;}

//var=70;
//aux=(long long)var*1000;
//if (aux==70000) output_toggle(led_verde2);
//delay_ms(20);
Leer_Linea();
prop=linea-1000;
//printf(" %ld %ld ",linea, prop);
if (prop<0) {output_high(led_rojo); output_low(led_verde);}
else {output_low(led_rojo); output_high(led_verde);}

#define umbral 150

if (abs(prop)<250)
{
output_high(led_verde2);
if (contrecta<umbral) contrecta++;
}
else
{
output_low(led_verde2);
}
}
}

```

```

        if (contrecta>0) contrecta--;
    }

    if (contrecta==umbral)
    {
        vel=1000;
        ult_prop=prop-ult_prop;
        //-----
        power_difference = (prop*18)/20 + ult_prop*15;
        //-----
        ult_prop=prop;
    }
    else

    {
        vel=800;
        ult_prop=prop-ult_prop;
        //-----
        power_difference = prop/20 + ult_prop*3;
        //-----
        ult_prop=prop;
    }

    }

// var = 50;
    if (power_difference > VEL) power_difference = VEL;

    if (power_difference < (-VEL)) power_difference = -VEL;

// printf(" %ld %ld %ld",linea, prop,power_difference);

    if(power_difference < 0)
        motores(VEL+power_difference, VEL);
    else motores(VEL, VEL-power_difference);

}

}

void test(void)
{
    int i;
    for(i=0;i<4;i++)
    {
        output_high(LED_ROJO);
        output_high(LED_VERDE);
        output_high(LED_AMARILLO);
        output_high(LED_VERDE2);
        delay_ms(100);
        output_low(LED_ROJO);
        output_low(LED_VERDE);
        output_low(LED_AMARILLO);
    }
}

```

```

    output_low(LED_VERDE2);
    delay_ms(100);
}
}

void calibrar(void)
{
int i;

for(i=1;i<8;i++)
{
    set_adc_channel(i);
    lectura[i]=read_adc();
    delay_us(50);
}

for(i=1;i<8;i++)
{
    if (lectura[i]<calibrado[i][0]) calibrado[i][0]=lectura[i];
    if (lectura[i]>calibrado[i][1]) calibrado[i][1]=lectura[i];
}
}

void calibracion(void)
{
int counter,i;

motores(0,0);
for(counter=0;counter<150;counter++)
{
    calibrar();
    delay_ms(20);
    output_toggle(LED_AMARILLO);
}
output_low(LED_AMARILLO);

printf("\n\r");
printf("\n\r");
for(i=1;i<3;i++)
{
    printf("%u: min %lu    max %lu ",i,calibrado[i][0],calibrado[i][1]);
}
printf("\n\r");

for(i=3;i<6;i++)
{
    printf("%u: min %lu max %lu ",i,calibrado[i][0],calibrado[i][1]);
}
printf("\n\r");
for(i=6;i<8;i++)
{
    printf("%u: min %lu max %lu ",i,calibrado[i][0],calibrado[i][1]);
}
}

```

```
}
```

```
void motores(long izq,long der)
```

```
{  
    set_pwm1_duty(izq);  
    set_pwm2_duty(der);  
}
```

```
void inicializar(void)
```

```
{  
    int i;  
    output_low(IN_M1);  
    output_low(IN_M2);  
    for(i=0;i<8;i++)  
    {  
        calibrado[i][0]=1023;  
        calibrado[i][1]=0;  
    }  
}
```

```
//-----
```

```
void Leer_Linea(void)
```

```
{  
    int i,j,enlinea;  
    signed long resta,dif;  
    signed long long cociente;
```

```
for(i=1;i<8;i++)
```

```
{  
    lectura[i]=0;  
}
```

```
for (j=0;j<5;j++)
```

```
{  
    for(i=1;i<8;i++)  
    {  
        set_adc_channel(i);  
        lectura[j]+=read_adc();  
        delay_us(50);  
    }  
}
```

```
for(i=1;i<8;i++)
```

```
{  
    lectura[i]=lectura[i]/5;  
}
```

```
printf("\n\r");
```

```
//for(i=1;i<8;i++) {printf("%u: %4lu ",i,lectura[i]);}
```

```

for(i=3;i<6;i++)
{
//printf("A%u: %4lu ",i,lectura[i]);
resta=calibrado[i][1]-calibrado[i][0];

}

enlinea=0;
for(i=3;i<6;i++) if (lectura[i]>200) enlinea=1;

if (enlinea==1)
{
cociente=(long long)lectura[4]*1000;
cociente+=(long long)lectura[5]*2000;
cociente=cociente/(lectura[3]+lectura[4]+lectura[5]);
linea=cociente;
}
else
{
if (linea<1000) { linea=0;}
else {linea=2000;}
}
// for(i=3;i<6;i++) printf("%u: %4ld ",i,lectura[i]);
//printf(" YYY: %4ld ",linea);
}

```