

Competencia de Robótica JOHNNY 5 2015

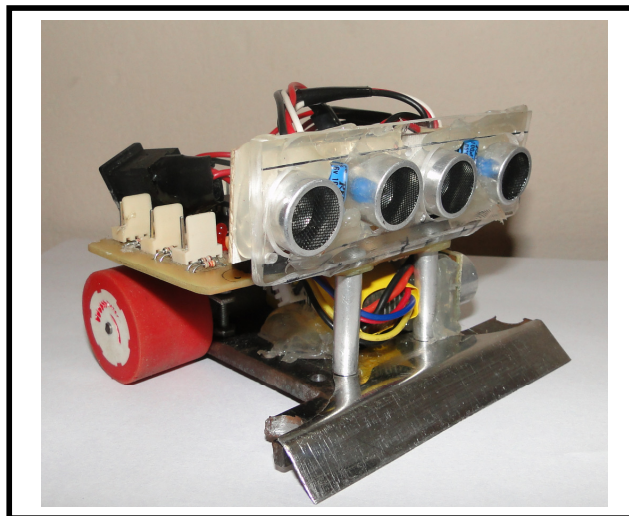
Categoría: MINISUMO

Nombre del Robot: CORAJE

Institución: EEST Nro. 6 SAN NICOLAS

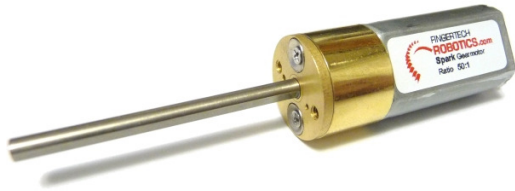
Participantes:

RAMIRO MARTIN PAEZ
ELIO ANTONIO PAEZ
RODRIGO NICOLAS PAEZ



El robot consta de los siguientes elementos que fueron comprados por internet(a un costado de cada imagen se encuentra el link).

Para la tracción dos motores de Fingertech, a los cuales se les corto el eje para poder acoplar las ruedas y estar dentro del reglamento de la categoría. Las ruedas también fueron compradas en el mismo lugar como las baterías de polímero de 360 mAh.



<http://www.fingertechrobotics.com/proddetail.php?prod=ft-spark16>

<http://www.fingertechrobotics.com/proddetail.php?prod=ft-spark16>

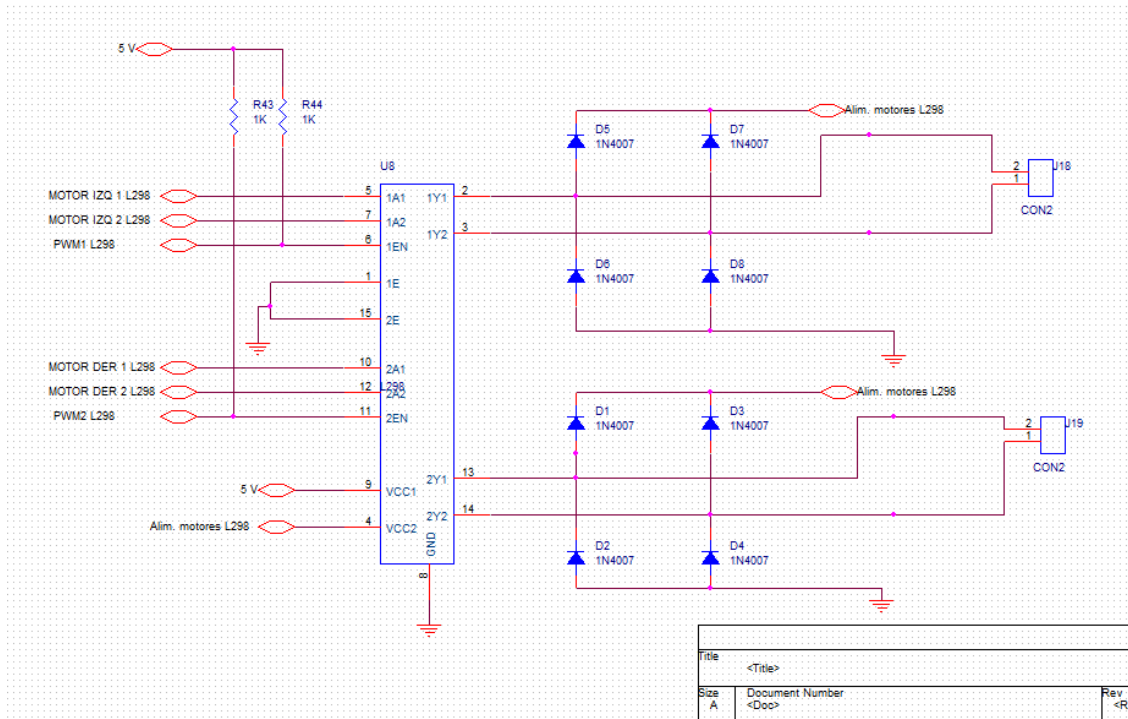


<http://www.fingertechrobotics.com/proddetail.php?prod=R-3S-mAh>

El diseño original era de cuatro ruedas por eso el chasis en forma de "T", pero debido a un "mal ensayo" tuvimos una rotura en uno de los motores. El chasis tiene la mayor parte del peso (igual estamos 100 gr. por debajo del reglamento). Posee una pala que intenta quitarle tracción al oponentes tratando de levantarlo. A los motores probamos pegarlos al chasis pero debido a los golpes tendía a despegarse y se decidió de anclarlos al chasis por medio de tornillos.

En lo que respecta a la electrónica utilizamos un pic 16F873A para la lógica y un L298 para el manejo de los motores, a este le agregamos todos los diodos de protección. Para la

Aca mostramos el puente H utilizado y sus conexiones.



Para la programación se utilizó MIKROBASIC, y el programa es el siguiente:

program FitoMinisumo

' Declarations section

```

symbol LED_ATRAS = portb.5      ' Led de indicacion de estado de sensores de ultrasonido
symbol LED_IZQ   = portb.6
symbol LED_ADEL  = portb.4
symbol LED_DER   = portb.7
" symbol PULS1   = porta.4      ' Declaracion pulsador de arranque de programa
" symbol PULS2   = porta.5
symbol PWMADELMI = portb.0      ' Habilitacion para que el motor izquierdo vaya hacia adelante
symbol PWMATMI   = portb.1      ' Habilitacion para que el motor izquierdo vaya hacia atras
symbol PWMADELMD = portb.2      ' Habilitacion para que el motor derecho vaya hacia adelante
symbol PWMATMD   = portb.3      ' Habilitacion para que el motor derecho vaya hacia atras

symbol CNYTRA    = porta.2      ' sensor de piso
symbol CNYIZQ    = porta.1      ' sensor de piso
symbol CNYDER    = porta.0      ' sensor de piso

symbol BUZZER    = portc.3
symbol ULTRALATDER = portc.4      ' sensor lateral del lado derecho
symbol FRENTEIZQ = portc.5      ' sensor delantero del lado izquierdo
symbol ULTRALATIZQ = portc.7      ' sensor lateral del lado izquierdo
symbol FRENTEDER = portc.6      ' sensor delantero del lado derecho

symbol ADELANTE  = portb = %00000101
symbol ATRAS     = portb = %00001010
symbol GIRODERECHA = portb = %00001001
symbol GIROIZQUIERDA = portb = %00000110

```

```

symbol derecha=1
symbol izquierda=0
"symbol GIROIZQARCOAD = porta = %0000

' Declaración de variables

dim aux as byte ' variable utilizada al comienzo para guardar la lectura inicial del objetivo
" dim intduty1 as byte ' variable utilizada para guardar el valor generado por la interrupcion de los
sensores de piso
" dim intduty2 as byte
" dim auxint as bit ' variable utilizada como bandera de las interrupciones, para evitar el problema
' del llamado de un procedimiento dentro de la misma.
dim current_duty1 as float
dim current_duty2 as float
dim contrutina as integer
dim contreinicio as integer

dim ultimomov,paso as byte "derecha=1 izquierda=0

dim temp_res as word "para la conversion de los CNY

sub procedure Configuracion
adcon1 = $04
porta = 0
trisa = %00001111
portb = 0
trisb = %00000000
portc = 0
trisc = %11110000

PWM1_Init(5000) ' Initialize PWM2 module at 5KHz
PWM2_Init(5000) ' Initialize PWM2 module at 5KHz

" auxint=0
Sound_Init(PORTC, 3)

PWM1_Set_Duty(0) ' Set current duty for PWM1
PWM2_Set_Duty(0) ' Set current duty for PWM2
PWM1_Start() ' start PWM1
PWM2_Start() ' start PWM2
end sub

sub procedure Tone1()
Sound_Play(659, 250) ' Frequency = 659Hz, duration = 250ms
end sub

sub procedure Tone2()
Sound_Play(698, 250) ' Frequency = 698Hz, duration = 250ms
end sub

sub procedure Tone3()
Sound_Play(784, 250) ' Frequency = 784Hz, duration = 250ms
end sub

```

```
sub procedure Tone4()
    Sound_Play(824, 250)      ' Frequency = 784Hz, duration = 250ms
end sub
```

```
sub procedure Tone5()
    Sound_Play(925, 250)      ' Frequency = 784Hz, duration = 250ms
end sub
```

```
sub procedure Melody()      ' Plays the melody "Yellow house"
    Tone1() Tone2() Tone3() Tone3()
    Tone1() Tone2() Tone3() Tone3()
    Tone1() Tone2() Tone3()
    Tone1() Tone2() Tone3() Tone3()
    Tone1() Tone2() Tone3()
    Tone3() Tone3() Tone2() Tone2() Tone1()
end sub
```

```
sub procedure Cincosegundos
    Tone1()
    delay_ms (750)      "delay del reglamento

    Tone2()
    delay_ms (750)      "delay del reglamento

    Tone3()
    delay_ms (750)      "delay del reglamento

    Tone4()
    delay_ms (750)      "delay del reglamento

    Tone5()
    delay_ms (750)      "delay del reglamento

    ultimomov = izquierda
end sub
```

```
sub procedure valorPWM

    PWM1_Set_Duty(current_duty1)
    PWM2_Set_Duty(current_duty2)

end sub
```

```
sub procedure probarmotores

    ADELANTE
    current_duty1 = 0      'este es el derecho
    current_duty2 = 255    'este el izquierdo
    valorPWM
    delay_ms (3000)
    ADELANTE
    current_duty1 = 255    'este es el izquierdo
    current_duty2 = 0      'este el derecho
    valorPWM
```

```

delay_ms (3000)
  ATRAS
  current_duty1 = 0   'este es el izquierdo
  current_duty2 = 255 'este el derecho
  valorPWM
  delay_ms (3000)
  ATRAS
  current_duty1 = 255 'este es el izquierdo
  current_duty2 = 0   'este el derecho
  valorPWM
  delay_ms (3000)
end sub

main:

Configuracion
Cincosegundos

while (1)

  aux= portc and %11110000
  select case aux
    case 0
      ADELANTE
      current_duty1 = 200
      current_duty2 = 0
      valorPWM

      " if ultimomov = derecha then
      "   ADELANTE
      "   current_duty1 = 255 'este es el derecho
      "   current_duty2 = 100 'este el izquierdo
      "   valorPWM
      " else

      "   ADELANTE
      "   current_duty1 = 100
      "   current_duty2 = 255
      "   valorPWM
      " end if
    case %01100000 'ambos sensores frontales detectando

      ADELANTE
      current_duty1 = 255
      current_duty2 = 255
      valorPWM

    case %00100000 'sensor frontal izquierdo detectando

      ADELANTE
      current_duty1 = 125
      current_duty2 = 0
      ultimomov= izquierda
      valorPWM

    case %01000000 'sensor frontal derecho detectando

```

```

        ADELANTE
        current_duty1 = 0
        current_duty2 = 125
        ultimomov=derecha
        valorPWM

    case %00010000      ' sensor lateral derecho detectando

        ADELANTE      'aca falta que gire rapido!!!!
        current_duty1 = 0  'y 90 grados lo tenemos que calcular
        current_duty2 = 250
        valorPWM

    case %10000000      ' sensor lateral izquierdo detectando

        ADELANTE      'aca falta que gire rapido!!!!
        current_duty1 = 250  'y 90 grados lo tenemos que calcular
        current_duty2 = 0
        valorPWM

    case else

        ' ruido asegurado

end select

"   temp_res = ADC_Read(0)   ' CNY DERECHO
"   if temp_res > 570 then
"       LED_DER=1
"       ATRAS
"       current_duty1 = 255
"       current_duty2 = 255
"       valorPWM
"   end if
"
"   temp_res = ADC_Read(1)   ' CNY IZQUIERDO
"   if temp_res > 570 then
"       LED_IZQ=1
"       ATRAS
"       current_duty1 = 255
"       current_duty2 = 255
"       valorPWM
"   end if
"   LED_IZQ=0
"   LED_DER=0
wend
end.

```

A este programa le falta todavía implementar el circuito de arranque, que esperamos implementar en esta semana.