

Proyecto 1

Fase 2: Parallax ActivityBot y Solución del Laberinto

ALGORITMOS Y ESTRUCTURA DE DATOS

SECCIÓN 20

MIÉRCOLES 07 DE SEPTIEMBRE

INTEGRANTES:

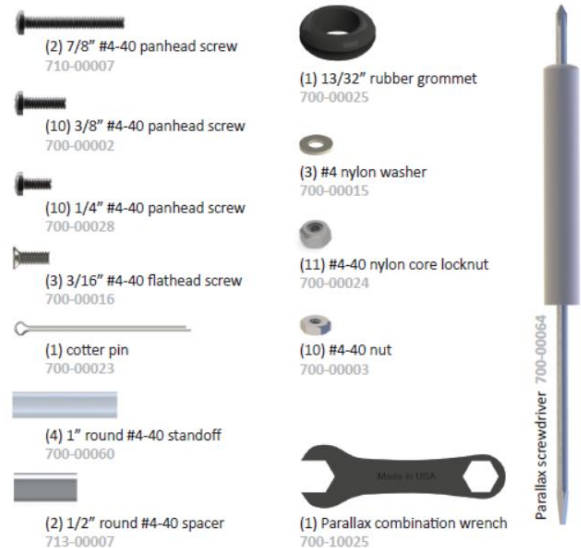
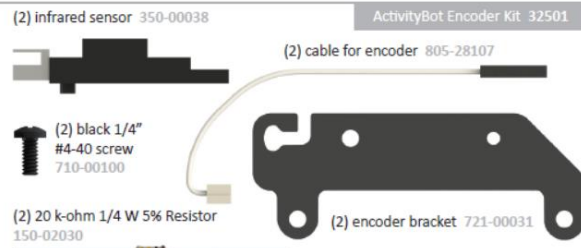
DIEGO VALDEZ CABRERA	15397
RAMÓN ANDRÉS SAMAYOA	15497
SANG WOO SHIN JI	15372
JORGE MARIO LARA	15366



1. Instructivo

I. En Caja:

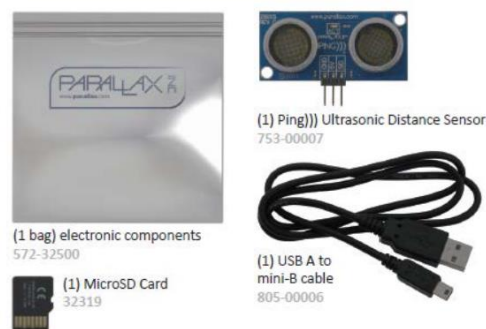
HARDWARE



You will have pieces left over. That's ok!



Additional parts (not used in this assembly guide)




Most, but not all, parts are sold separately from the web store. Contact Parallax Sales directly for any part: sales@parallax.com, 888-512-1024 in U.S., or 916-624-8333.

II. Ensamblaje del ActivityBot:

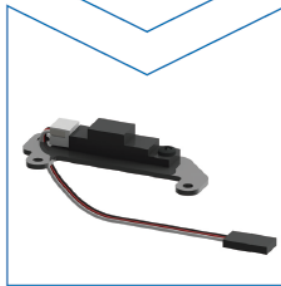
1) Preparar los Encoders

STEP 1

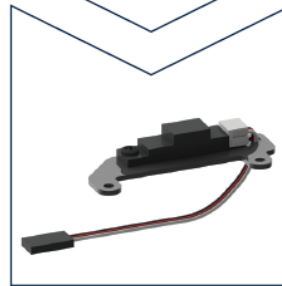
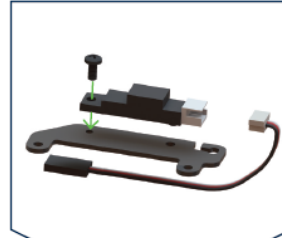
Prepare right and left encoders.

 (2) black 1/4" screw 710-00100

RIGHT



LEFT



2) Preparar las Llantas

STEP 2

Push a tire onto each wheel. Repeat.

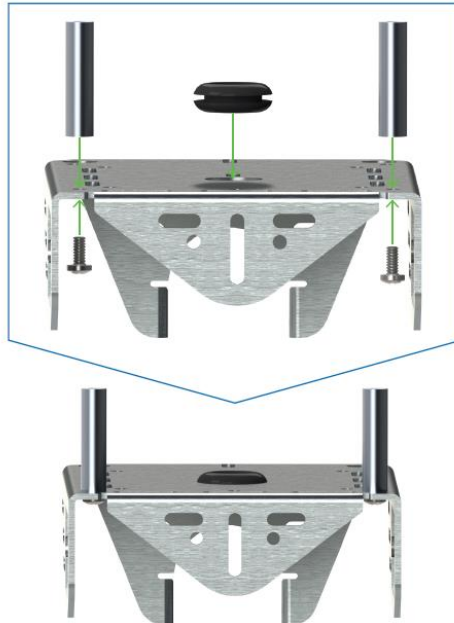


3) Preparar el Chasis

STEP 3

Prepare chassis.

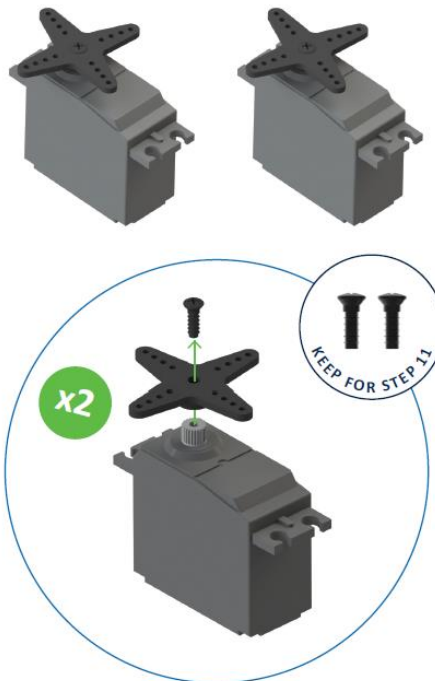
(2) 1/4" panhead screw 700-00028 (2) 1" round standoff 700-00060



4) Preparar los Servos

STEP 4

Remove servo horns. *Save the screws!*

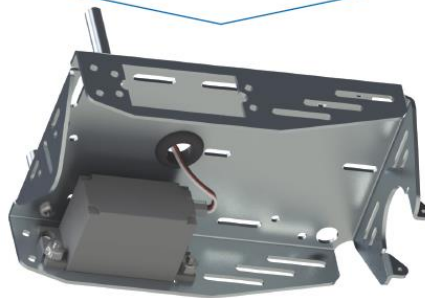
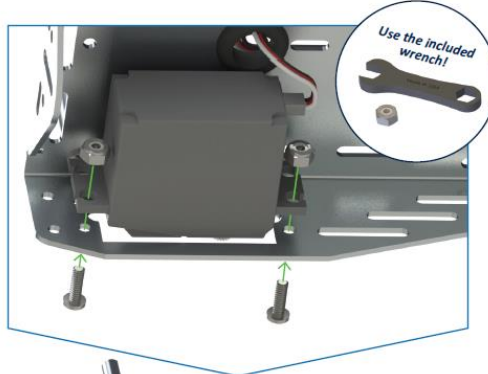


5) Montar el Servo Derecho

STEP 5

Mount right servo.

(2) 3/8" panhead screw 700-00002 (2) nylon core locknut 700-00024

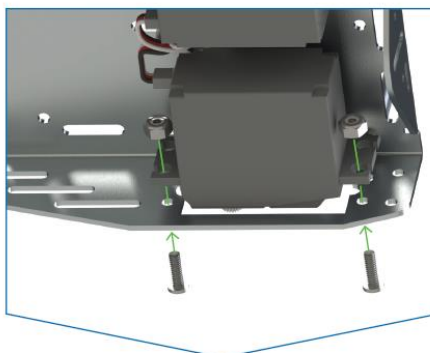


6) Montar el Servo Izquierdo

STEP 6

Mount left servo.

(2) 3/8" panhead screw 700-00002 (2) nylon core locknut 700-00024

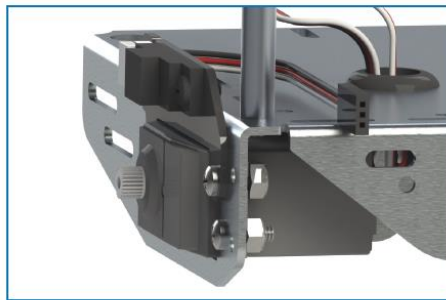
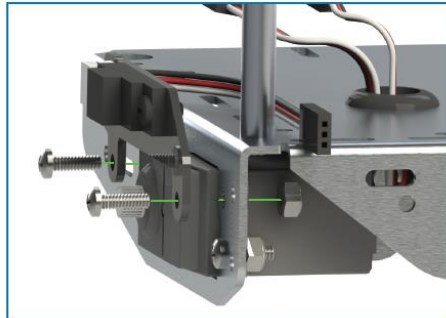


7) Montar el Encoder Derecho

STEP 7

Mount right encoder.

(2) 3/8" panhead screw 700-00002 (2) nylon core locknut 700-00024

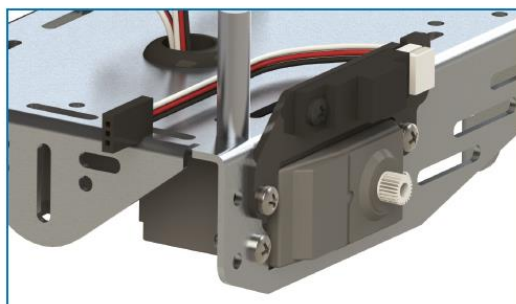
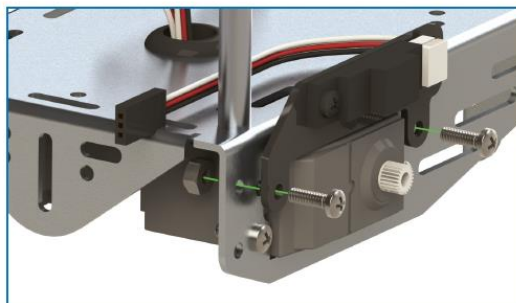


8) Montar el Encoder Izquierdo

STEP 8

Mount left encoder.

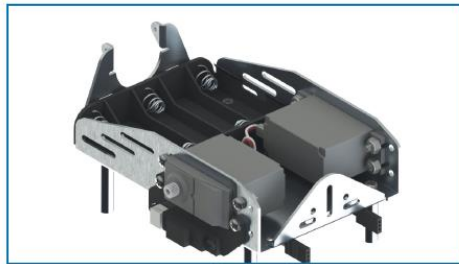
(2) 3/8" panhead screw 700-00002 (2) nylon core locknut 700-00024



9) Montar el Pack de Baterías

STEP 9 Mount battery pack.

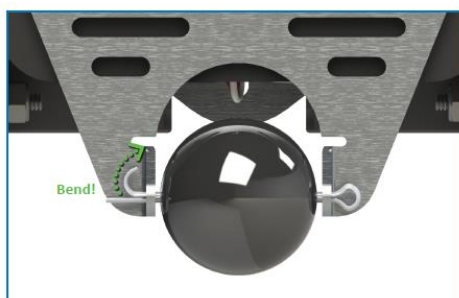
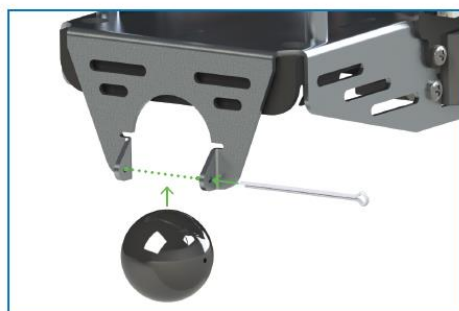
(2) 3/16" flathead screw 700-00016 (2) 1" round standoff 700-00060



10) Montar la Tail Wheel

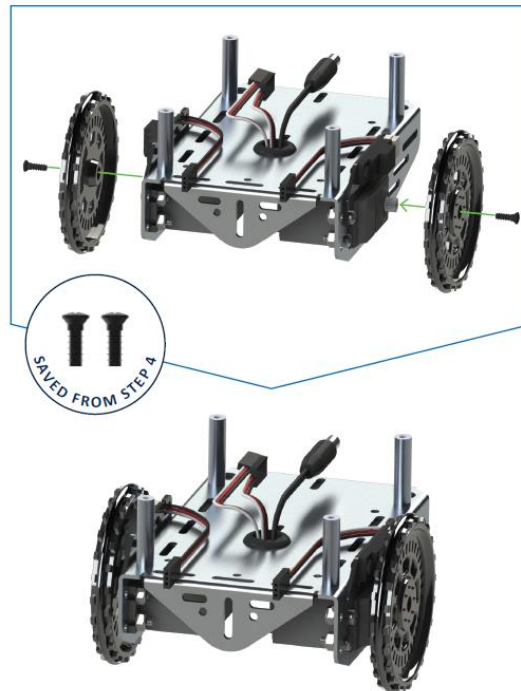
STEP 10 Mount tail wheel.

(1) cotter pin 700-00023



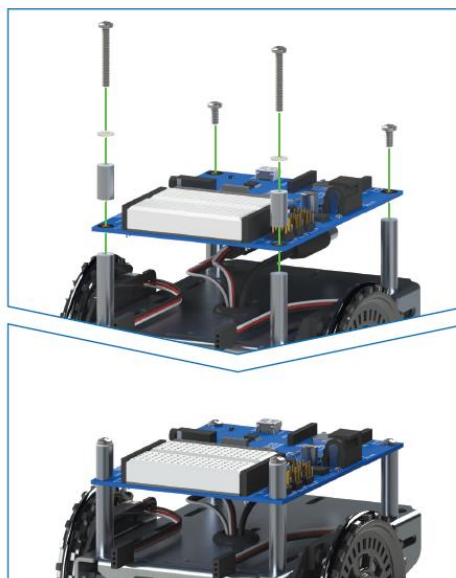
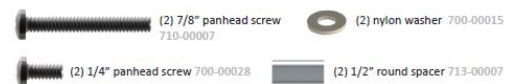
11) Montar las Llantas

STEP 11 Mount drive wheels.



12) Montar la Activity Board

STEP 12 Mount board.



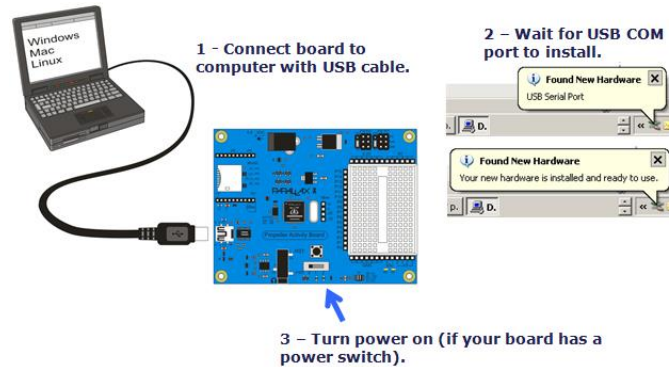
III. Software Necesario:

Solo es necesario de instalar Propelle C – SimpleIDE de parte de Parallax en el sistema de preferida elección. Se puede encontrar en el siguiente link:

<http://learn.parallax.com/tutorials/language/propeller-c/propeller-c-set-simpleide>

Para en el caso más común, WINDOWS, se realizaría de la siguiente manera:

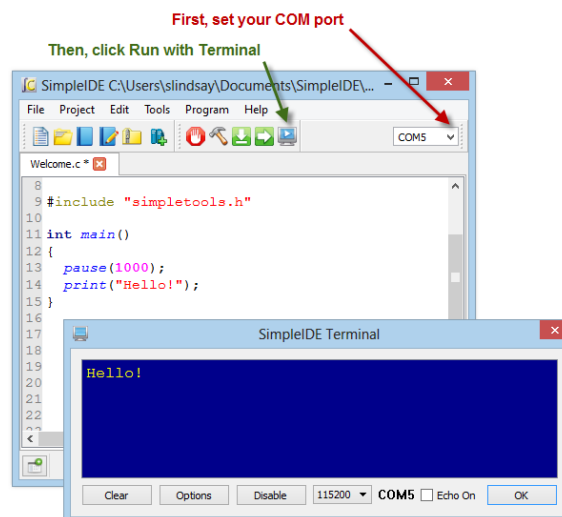
- Correr el instalador de SimpleIDE.
- Conectar el cable USB a la Propeller Activity Board.



- Colocar el Switch a la segunda posición (1) de 'Set Up', y esperar que el puerto USB termine de configurar los Drivers necesarios.



- Seleccionar el puerto correcto donde se tiene conectado el cable USB.
- Desde aquí ya se puede Correr en la Terminal los Códigos.



IV. Información de Hardware y su configuración:

1) Sensor Ultrasónico

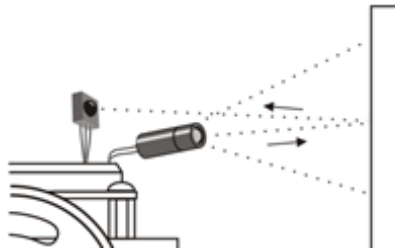


Este sensor PING))), genera un sonido (clicks) los cuales rebotan en superficies perpendiculares y regresa al sensor, con ello el tiempo en el que se tarda en ir y regresar la señal, a partir de esos datos saca la distancia a la que está un objeto de él, logra leer hasta distancias de 3 metros.

Configuración:

```
DistVariable = ping_cm(#pin)
```

2) Sensor Infrarrojo



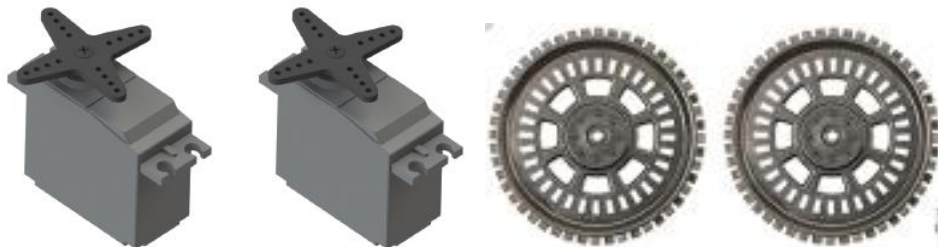
Estos son sensores los cuales funcionan a base de luz a cierta frecuencia. Este funciona con un emisor y un receptor, este solo te dice si un objeto está cerca o no, no especifica la distancia, por lo que solo muestra un 1 o un 0.

Configuración:

```
freqout(1, 1, 28000) //Frecuencia del LED
```

```
IrSensorVariable = input(#pin)
```

3) Servomotores (Llantas)



Estos son motores giratorios los cuales son controlados con señales eléctricas para la velocidad a los que estos giran, no tienen restricción de ángulo por lo que son 360, lo que quiere decir que puede realizar vueltas completas.

Configuración:

```
drive_speed(Wheel1, Wheel2)
```

2. Desempeño del Algoritmo

El algoritmo utilizado en el robot para la fase final fue el algoritmo de la mano derecha, el cual consiste en recorrer el laberinto siguiendo la pared derecha y este llevara a la salida del mismo. El algoritmo que se planeaba utilizar era un algoritmo que utilizaba recursividad para resolverlo, haciendo que este se llamara al mismo cuando encontraba varias posibilidades de caminos.

El algoritmo de la primera entrega con el de la segunda no era tan eficiente en el robot que nosotros teníamos por el problema de que este tenía un desperfecto mecanice y este no hubiera podido parar leer y luego moverse, el algoritmo para la entre final fue más eficiente ya que este se componía solo y este no paraba a analizar si no que este iba analizando mientras recorría el laberinto.

El algoritmo consistía en que este con el sensor infrarrojo del lado derecho mientras hubiera una pared del lado derecho este seguía caminando, cuando no encontrara una pared del lado derecho este hacia un giro de 90 hacia la derecha, y este también giraba hacia la izquierda cuando encontraba pared enfrente y a la derecha al mismo tiempo, con esto se habría terminado el algoritmo, pero al tener problemas de que este se desviara se agregó que este al topar de frente de algo y no avanzar este retrocedía y giraba hacia la izquierda para seguir y detectar la pared derecha.

El robot hacia un recorrido de modo continuo ya que este tenía un error que era que si este se detenía aunque uno le pusiera velocidad 0 un motor no paraba y este se desviaba por eso se realizó el algoritmo para que este no tuviera que parar en ningún momento.

LINK A GITHUB:

<https://github.com/Galos96/LaberintoRobot>