

**Universidad del Valle de Guatemala**

**Campus Central**

**Facultad de Ingeniería**



**Algoritmos y Estructura de Datos**

## **PROYECTO ROBOT QUE SALE DE UN LABERINTO**

**Realizado por:**

Gladys De La Roca 15755

Jackeline Hidalgo 15776

Andrea Maybell Peña 15127

Steven Rubio 15044

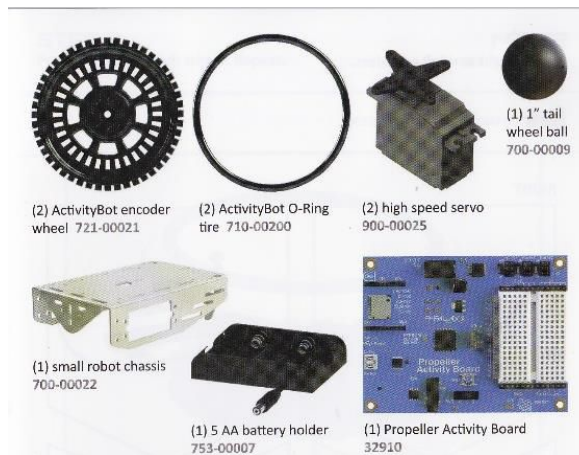
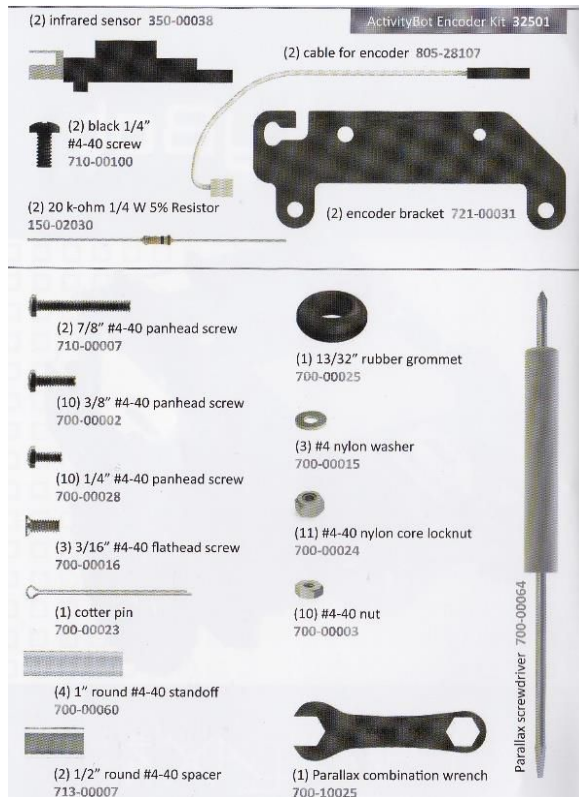
**Catedrático: Lynett Garcia**

**Sección: 10**

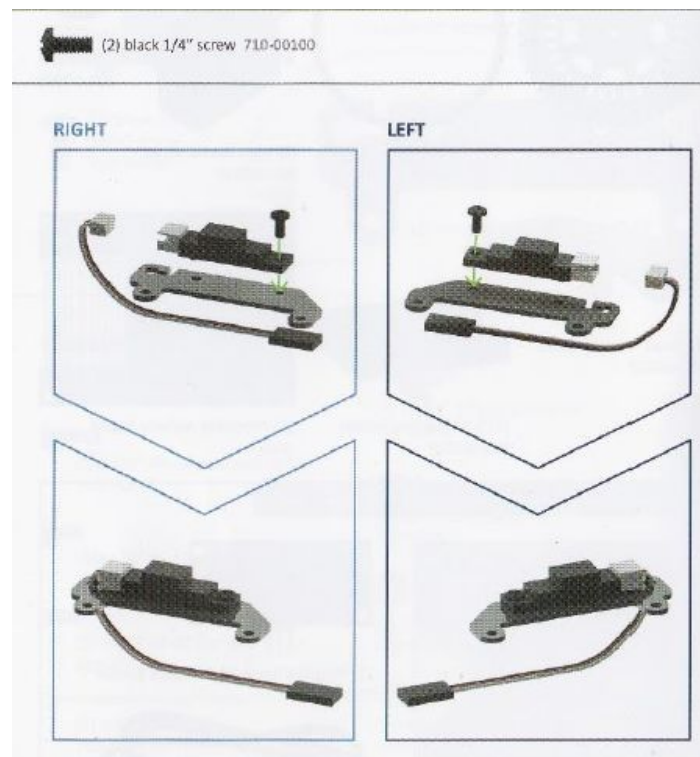
**Guatemala, 7 de septiembre del 2016**

# I. Ensamblaje del Chasis

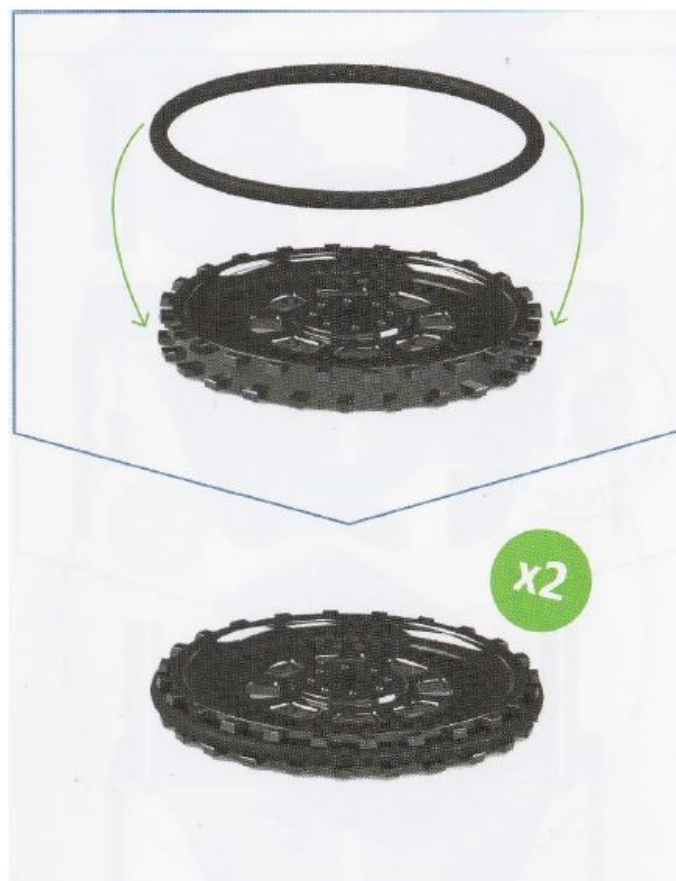
Piezas:



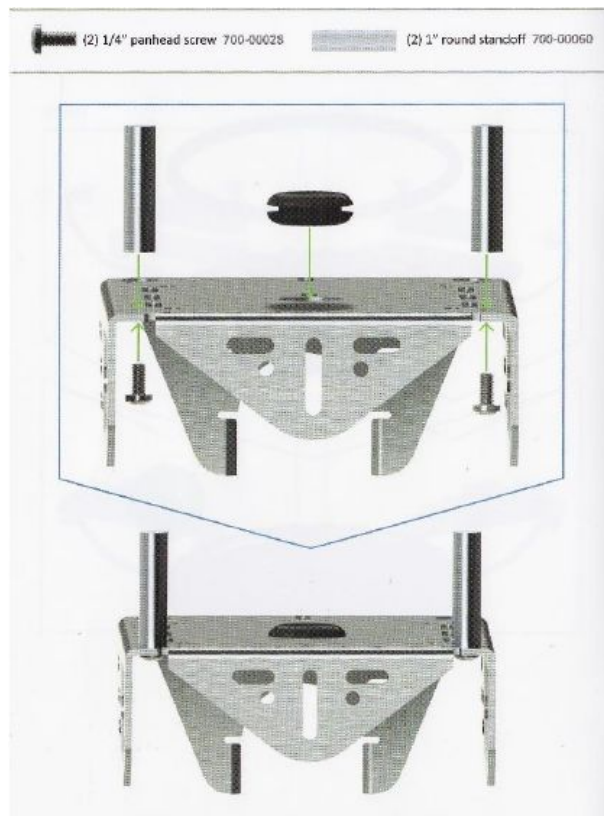
## 1. Preparar Encoders



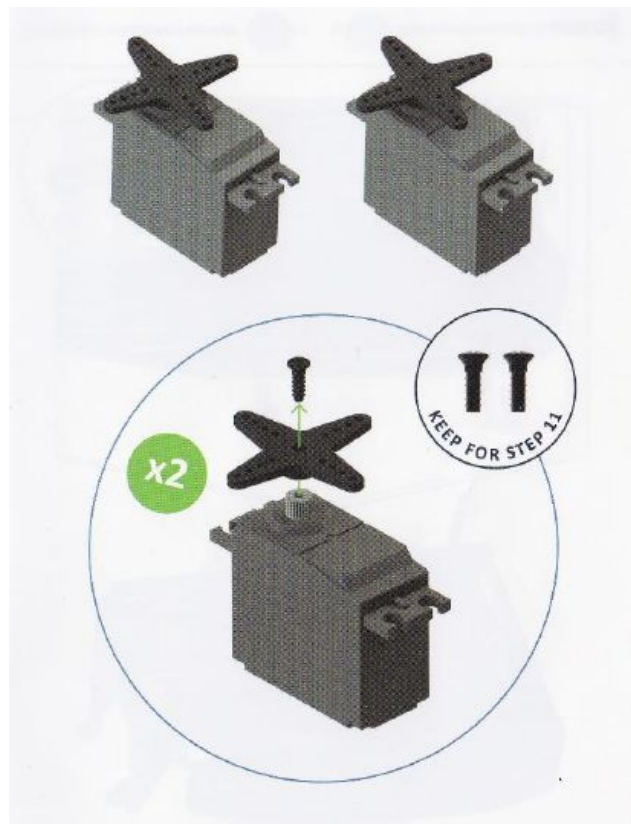
## 2. Preparar Ruedas



### 3. Armar el chasis



### 4. Preparar servomotores para su instalación

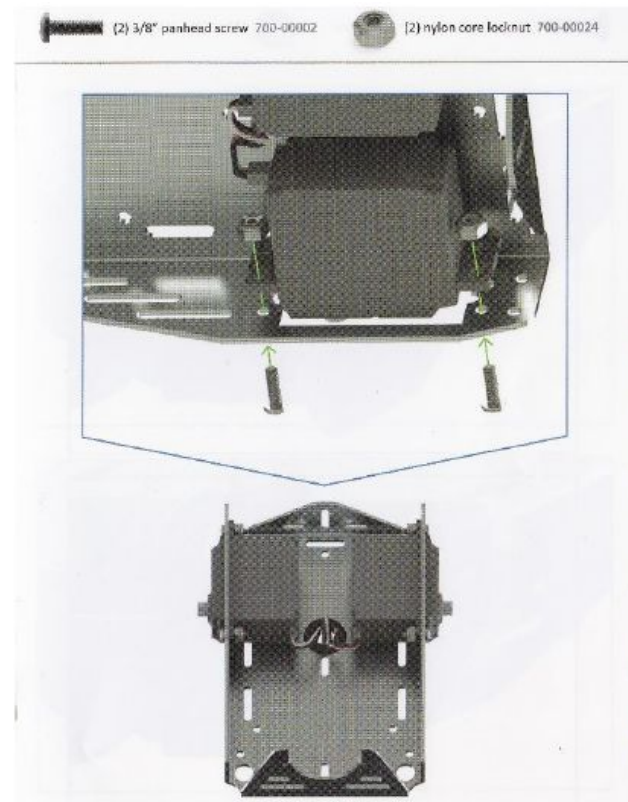




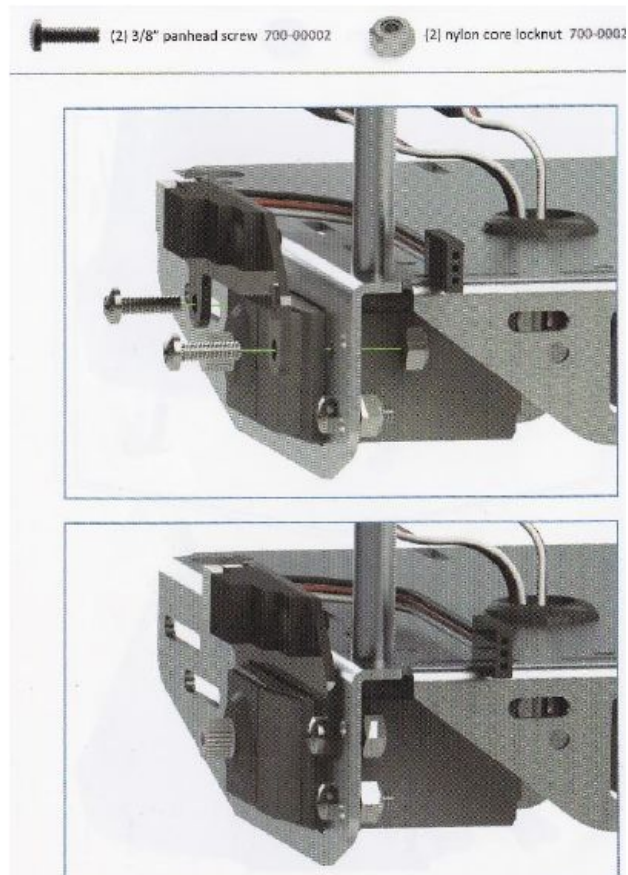
5. Montar el servo del lado derecho con ayuda del destornillador y la pequeña llave plástica incluida



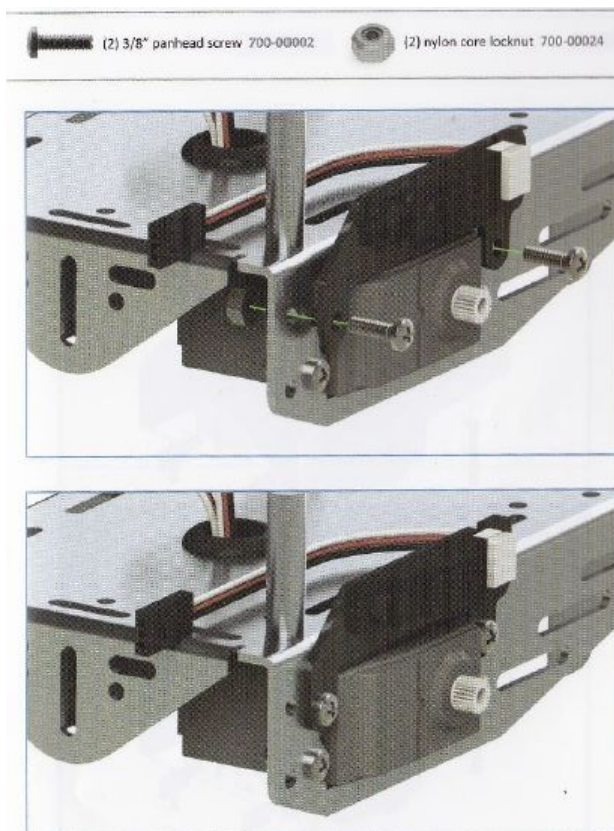
6. Montar el servo del lado izquierdo con ayuda del destornillador y la pequeña llave plástica incluida



## 7. Montar encoder derecho



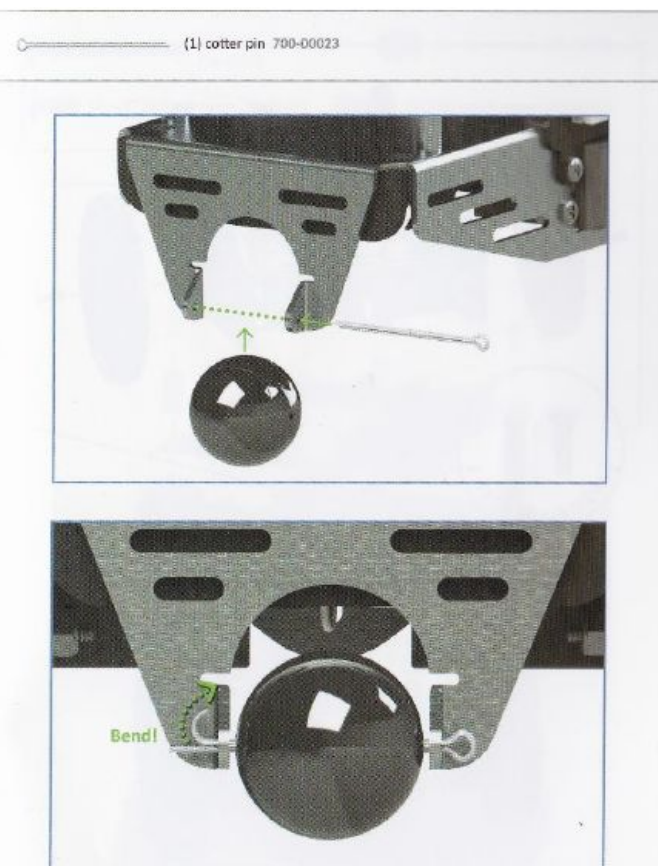
## 8. Montar encoder izquierdo



## 9. Colocar adaptador de baterías

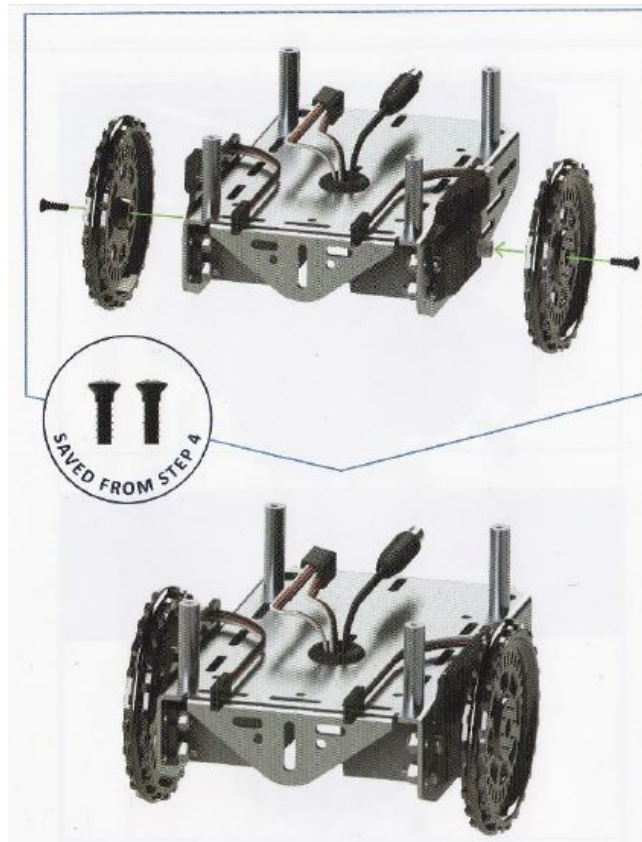


## 10. Montar rueda independiente





## 11. Montar ruedas en los servos



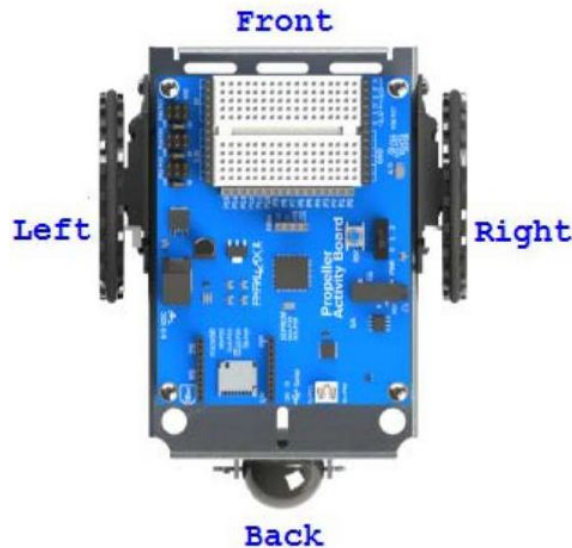
## 12. Montar la placa en el chasis





### Conexiones Eléctricas:

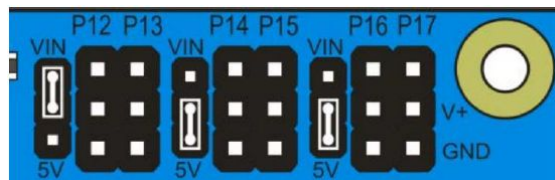
Tomar en cuenta la posición del robot.



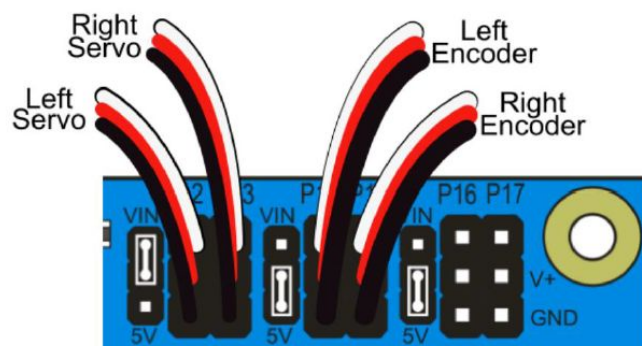
### Preparando puertos del servo:

Cada conector de la tarjeta cuenta con un jumper, es importante que al manipular esto jumpers no tenga ningún tipo de energía la tarjeta. Los jumpers deben de estar colocados como se muestra a continuación.

**IMPORTANTE:** Asegurarse que los jumpers estén bien colocados, una mala ubicación puede dañar los servos o la placa.

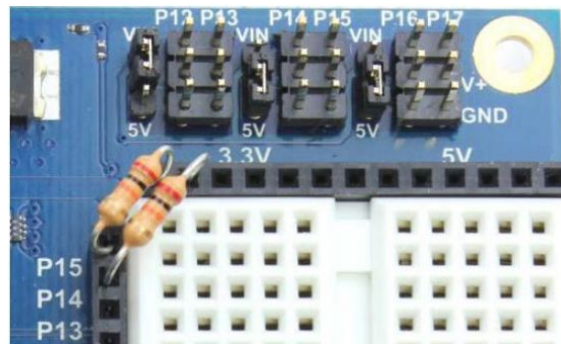


Cada conector cuenta con tres colores de cables (blanco, rojo y negro). Los cables blancos deben de estar en el extremos más cercano al borde de la tarjeta. Mientras el negro debe de estar conectado a tierra. El esquema de la conexión de los cables se muestra a continuación.



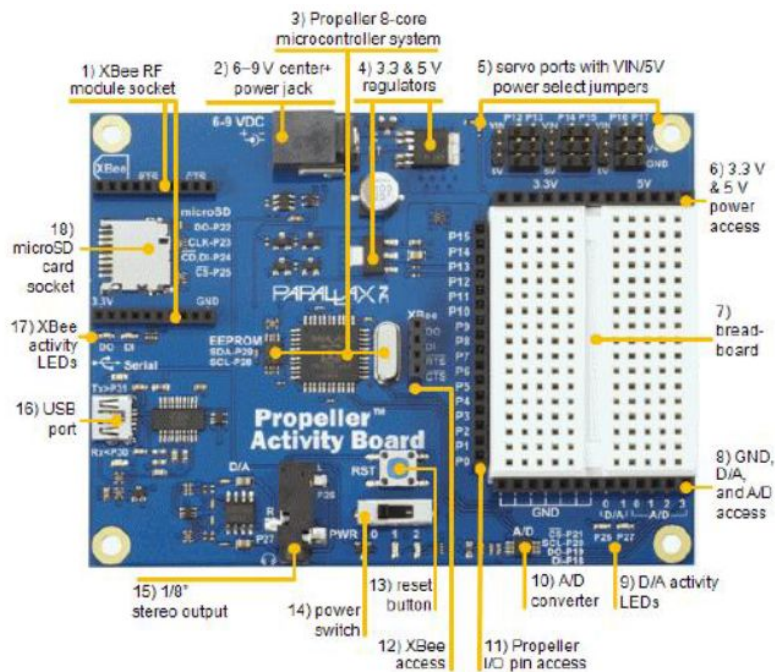
Agregar resistencias:

Para seguridad de los componentes se colocarán dos resistencias de 20k-ohms en los puertos 14 y 15 de protoboard que se encuentra en la tarjeta, el otro extremo se conectará a la salida de 3.3V. El esquema se muestra a continuación.



Partes de la placa:

A continuación se muestra señaladas las partes de la placa con la que cuenta el ActivityBot.



Posiciones del Switch:

La placa cuenta con un switch de tres posiciones, cada uno con una función en específico, a continuación:

Posición 0: apagado total.

Posición 1: Carga del programa, únicamente se provee energía a la placa.

Posición 2: se provee energía a todos los componentes de la placa, los servos y Encoders. Esta posición sólo debe de habilitarse en el momento en que se desea correr el programa. De no ser así, el ActivityBot puede moverse en una superficie inestable y/o caer de donde esté colocado. Previo a mover el switch a esta posición se debe presionar el botón de reset de la placa.

## II. Instalación del Software

El ActivityBot se programa en el lenguaje de alto nivel C. Para la programación se utilizará el software de SimpleIDE. Este está disponible para Windows, Mac , Linux y Raspberry.

En el siguiente link se puede encontrar las descargas e instrucciones para instalación de los drivers: <http://learn.parallax.com/propeller-c-set-simpleide>.

### INSTALACIÓN DE LA LIBRERÍA DE ACTIVITYBOT

1. Descargar de:

<http://learn.parallax.com/sites/default/files/content/propeller-c-tutorials/ActivityBot/Software/ActivityBot%202013-10-31.zip>

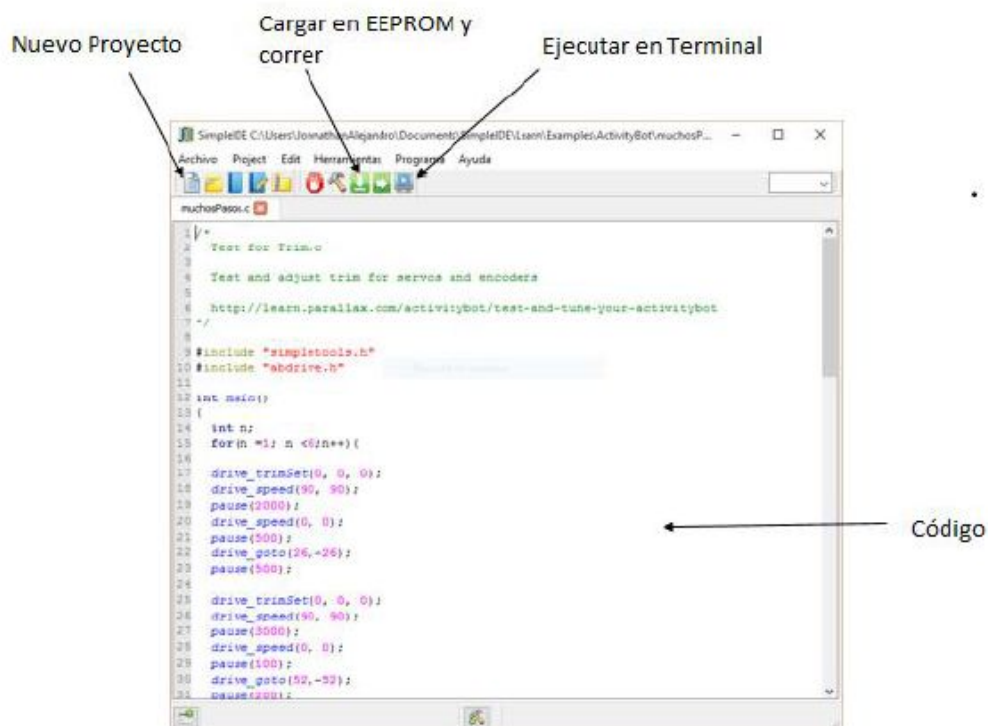
2. Descomprimir la carpeta dentro del zip.

3. Copiar la carpeta en la ruta de instalación del SimpleIDE (Documents\SimpleIDE\Learn\Simple)

Importante: si cuenta con una versión previa de esta librería debe de eliminarla. Para que los cambios sean efectivos debe de reiniciar el programa.

## III. Ambiente de Programación

Interfaz del programa:



## IV. Calibración del Robot

Antes de ejecutar cualquier otro programa de ejemplo, su ActivityBot necesita ser calibrado . Se trata de una sola vez calibración que necesita la biblioteca abdrive para medir y corregir distancias y velocidades , con la información de los encoders del ActivityBot .

La calibración recolecta la velocidad requerida frente a los datos medidos de la velocidad y lo almacena en una parte de la memoria EEPROM del ActivityBoard, donde puede retener datos incluso después de que se desconecta el aparato. De esa manera, los datos de calibración está disponible cada vez que encienda el robot de nuevo.

Cuando el programa pide una cierta velocidad de las ruedas , la biblioteca abdrive utilizará los datos de calibración de la EEPROM para iniciar la conducción de los motores a velocidades cercanas a lo que su programa pide. De esa manera, abdrive no tiene que hacer grandes correcciones, lo que mejora la precisión global .

## V. Sensores y Circuito

ActivityBot, cuenta con dos sensores, a continuación le mostraremos cómo conectar dos de ellos.

Sensor ultrasónico: un sensor ultrasónico, es un sensor de proximidad que trabajan libres de roces mecánicos y pueden detectar distancia de hasta 8 metros. El sensor emite un sonido y este mide el tiempo en que la señal tarda en regresar para poder calcular la distancia en la que el obstáculo se encuentra o si este no existe.

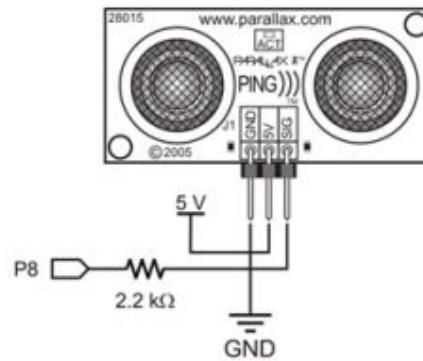
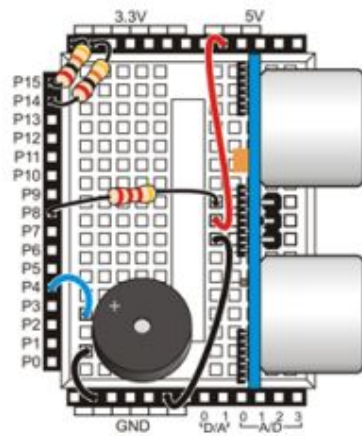


circuito:

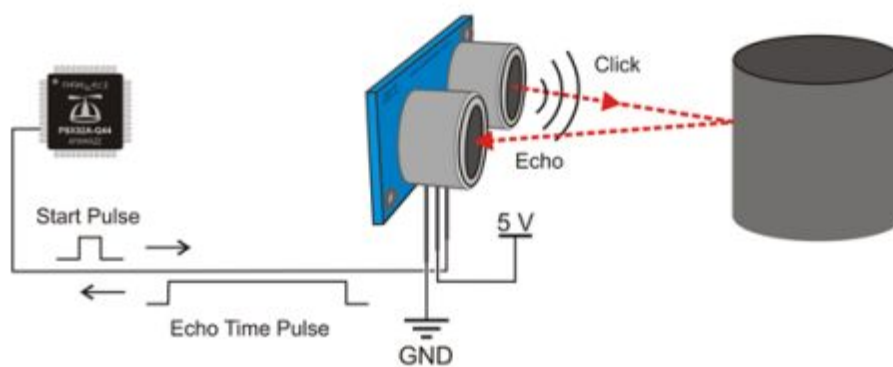
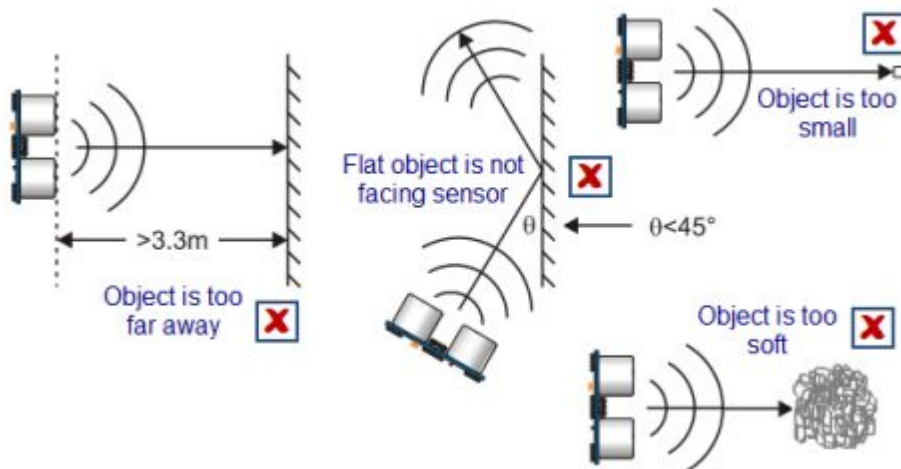
materiales:

- (1) PING))) Sensor (mostrado arriba)
- (1) Resistencia 2.2K ohm (rojo-rojo-rojo)
- (1) Jumper





Funcionamiento:



Código de prueba:

```
/*
  Test Ping.c
  Test the PING))) sensor before using it to navigate with the ActivityBot.
*/

#include "simpletools.h"           // Include simpletools header
#include "ping.h"                 // Include ping header

int distance;                    // Declare distance variable

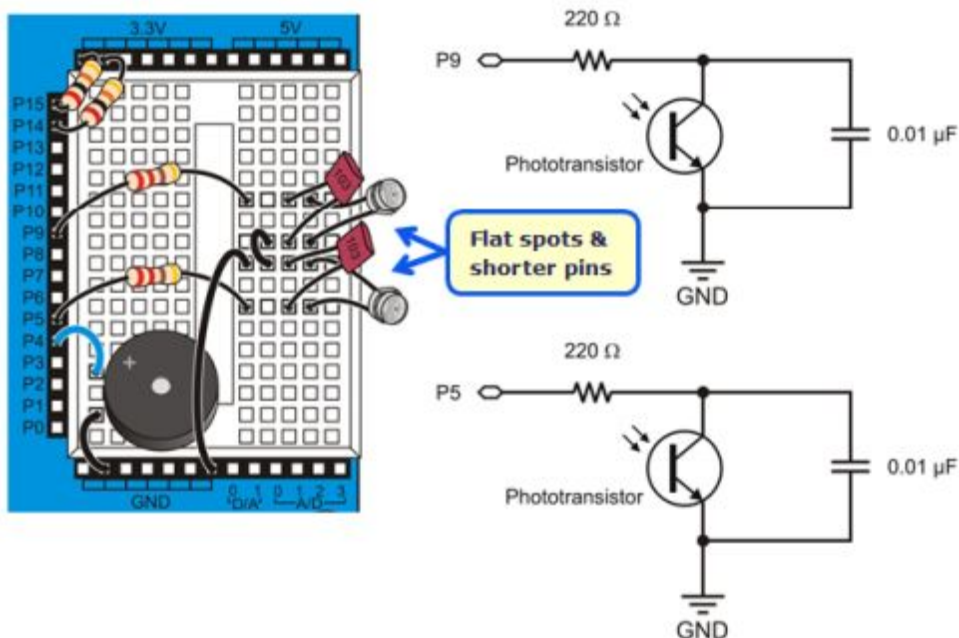
int main() {                     // main function
  while(1) {                     // Repeat indefinitely
    distancia = ping_cm(8);       // Get distance in cm from Ping)))
    print("%c distance = %d cm", HOME, distance, CLREOL); // Display distance
    pause(200);                  // Wait 1/5 second
  }
}
```

Sensor infrarrojo: un sensor infrarrojo es un dispositivo optoelectrónico capaz de medir la radiación proveniente del reflejo de la luz emitida por un LED especial, en este caso el sensor infrarrojo puede medir distancias de hasta 30 centímetros.

Circuito:

materiales:

- (2) Fotorresistencia (#350-00029)
- (2) Capacitor 0.01  $\mu\text{F}$  (103)
- (2) Resistencias 220 ohm (rojo-rojo-café)



Funcionamiento:



Código de prueba:

```
/*
 * Test Ping.c
 * Test the PING))) sensor before using it to navigate with the ActivityBot.
 */

#include "simpletools.h"           // Include simpletools header

int lightLeft, lightRight;       // Declare distance lightLeft, lightRight

int main() {                     // main function
    while(1) {                   // Repeat indefinitely
        high(9);
        pause(1);
        lightLeft = rc_time(9,1)

        high(5);
        pause(1);
        lightRight = rc_time(5, 1);

        print("%c distance = %d, lightRight = %d%c",
              HOME, lightLeft, lightRight, CLREOL); // Display distance

        pause(200);              // Wait 1/5 second
    }
}
```

## VI. Robot sale del Laberinto

En la realización de la primera fase de este proyecto se implementó el algoritmo de la mano derecha.

Es un algoritmo muy sencillo y ha sido utilizado desde hace cientos de años para encontrar salidas de laberintos, catacumbas o galerías subterráneas.

Este consiste en que hagamos uso de la mano derecha y la coloquemos sobre alguna pared y de esta manera recorrer el laberinto sin despegar la mano para asegurarnos que pasamos por todas las paredes. Eventualmente este nos llevara a la salida.

Para la segunda fase del proyecto, la implementación del laberinto en el robot, se utilizó el mismo algoritmo, como se podrá observar en el funcionamiento del mismo,

este evalúa siempre su derecha y de no poder girar a la derecha evalúa al frente, si este no tiene opción enfrente finalmente gira hacia la izquierda.

Se le agregó una función extra, la cual el robot evalúa la distancia disponible hacia el frente, si esta es mayor o igual a 50 cm el robot recorre los 50 cm sin evaluar su derecha.

link: <https://youtu.be/KuNDzJZf6-M>

## **VII. Referencias**

Imágenes e información obtenida en:

Parallax, Inc. (s.f.). <http://media.digikey.com/>. Consulta: 28 de agosto de 2016 en

<http://learn.parallax.com>:

<http://media.digikey.com/pdf/Data%20Sheets/Parallax%20PDFs/ActivityBot.pdf>