

UNIVERSIDAD DEL VALLE DE GUATEMALA

FACULTAD DE INGENIERIA



## PARALLAX ACTIVITYBOT

Diseño de un algoritmo para salir de cualquier laberinto

**Algoritmos y Estructuras de Datos, Sección 30**

Por: Jonnathan Juárez, carné 15377

Diego Castañeda, carné 15151

Javier Jo, carné 14343

2016

## 1. Ensamblaje

Para ensamblar el activitybot puede consultar directamente el manual provisto dentro de la caja del activitybot o dirigirse al siguiente enlace:

<http://learn.parallax.com/activitybot/mechanical-assembly>

Para la implementación de este proyecto el activitybot se encontraba armado por lo que no fue necesario realizar este paso.

## 2. Conexiones Eléctricas

Para realizar las conexiones eléctricas del activitybot se deberá contar con el robot ya armado, dos resistencias de 20K-ohm (rojo-negro-anaranjado) y cinco baterías AA.

Previo a realizar las conexiones por seguridad se debe de retirar cualquier tipo de suministro de energía, ya sea las baterías (en caso las tuviese) o el cable USB. Se deben conectar los cables de los servos y los sensores de las ruedas del robot en la siguiente configuración:

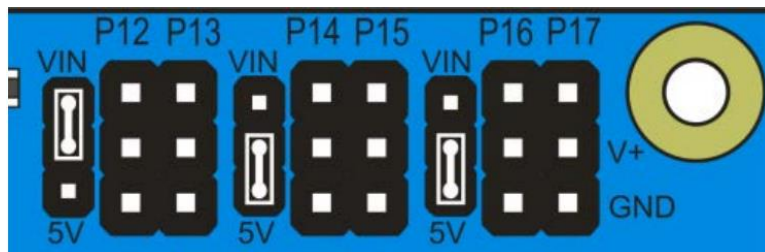
P12: servo izquierdo

P13: servo derecho

P14: sensor izquierdo

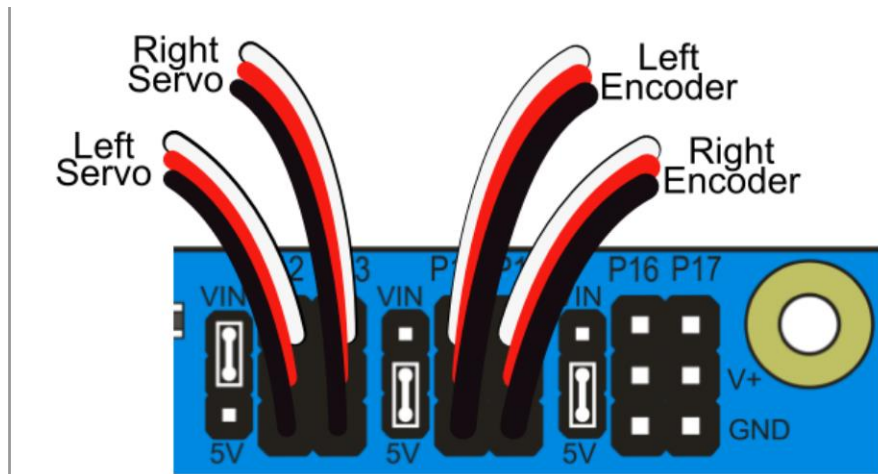
P15: sensor derecho

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.

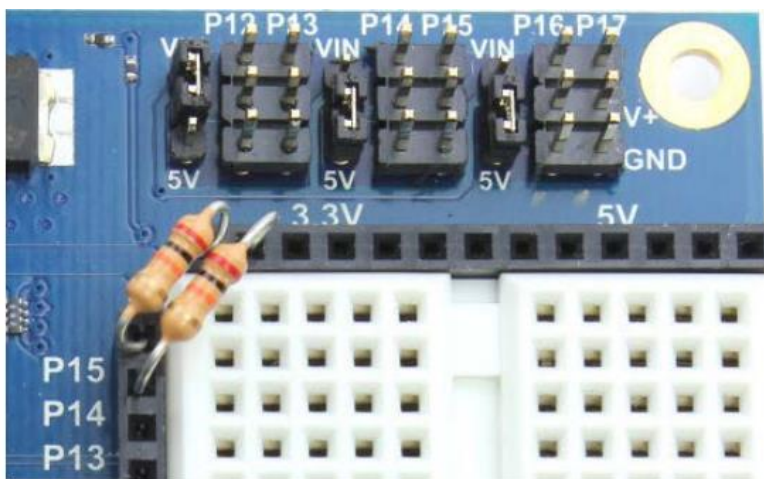


COLOCAR LOS CABLES:

Cada conector cuenta con tres colores de cables (blanco, rojo y negro). Los cables blancos deben de estar en el extremo 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.



Finalmente las resistencias de 20K-ohms. Estas deben ir conectadas del pin 14 y pin 15 a los puertos de 3.3V como se muestra en la figura.



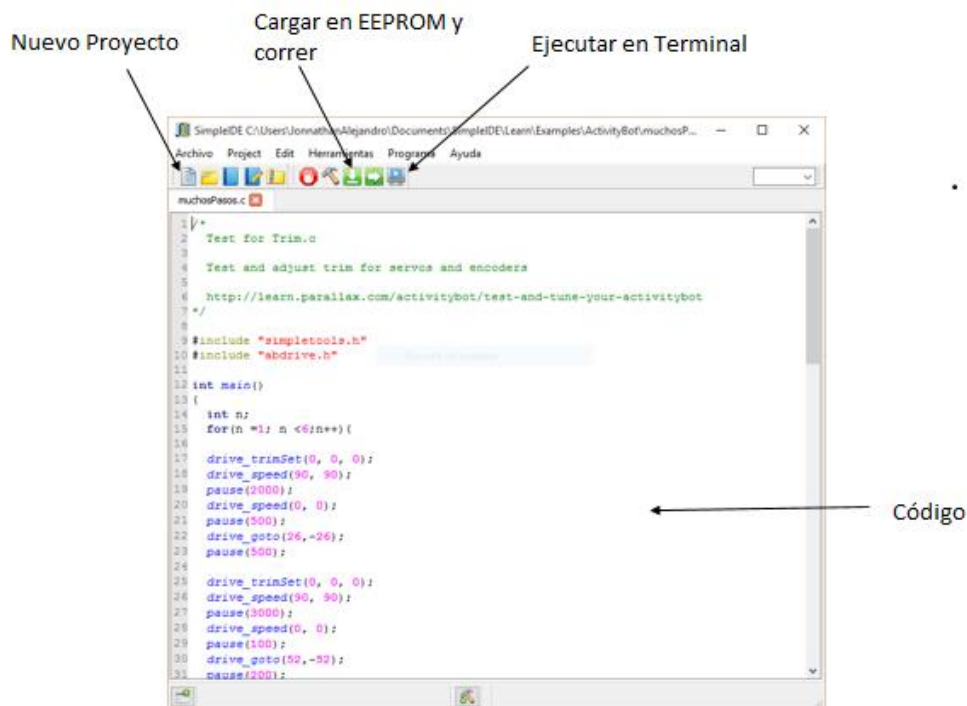
### 3. Instalando el Software

Se procederá a instalar el software SimpleIDE. Este herramienta ayuda a comunicarse con el robot así como a compilar los proyectos que se desarrollen en lenguaje C. Además proporciona librerías especializadas para trabajar con la placa de Parallax-

Para instalar el software diríjase al siguiente enlace y seleccione su sistema operativo.

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

Luego seleccione la opción de descargar y siga las instrucciones del instalador. Para un funcionamiento óptimo se recomienda utilizar la configuración estándar. Finalmente se recomienda instalar un driver extra (Windows solo) para mejorar la estabilidad de comunicación con el ActivityBot.



## Programación en SimpleIDE

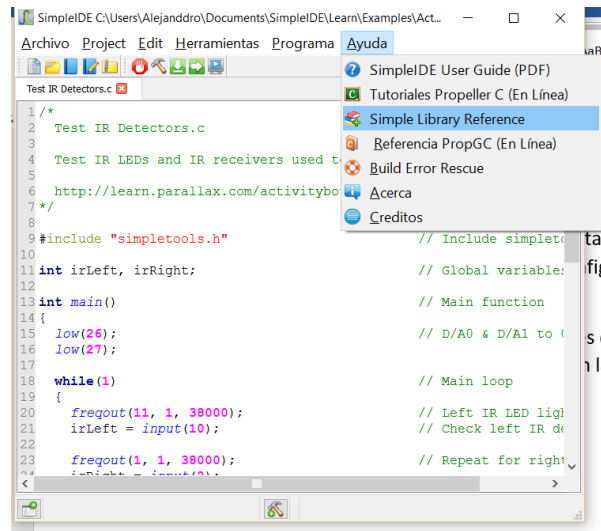
Para realizar cualquier código que necesite de uso de los sensores o servos del activitybot se deben de importar las siguientes librerías.

```
#include "simpletools.h"  
#include "abdrive.h"
```

Habiendo importado las bibliotecas usted podrá utilizar las funciones del activitybot. Además de contener las funciones del activitybot estas bibliotecas incluyen otras bibliotecas propias de C.

Es de recalcar que los servos con los que cuenta el robot no pueden girar a más de 120rpm por lo tanto en las funciones en las que se configuran velocidades de desplazamiento, no se debe de exceder este número.

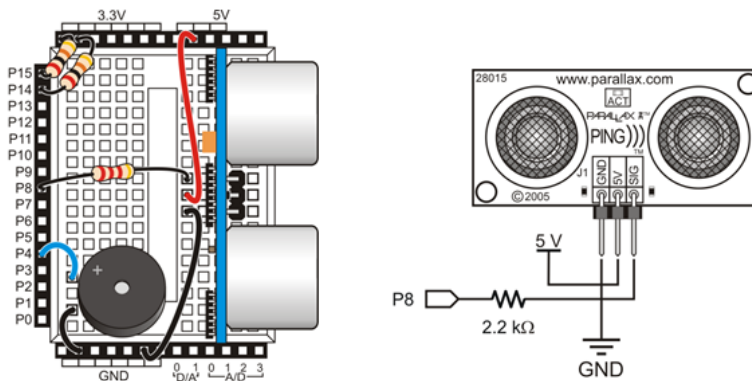
Para obtener más información de las funciones del robot puede visitar la parte de ayuda> Simple Library Reference. que se encuentra en la barra de menús de SimpleIDE.



## Sensores:

Para ayudar al robot a navegar entre los distintos tipos de obstáculos, se pueden utilizar varios tipos de sensores, sin embargo, en esta implementación se utilizó un sensor ultrasónico que ayuda a medir la distancia frente a él y los infrarrojos que detectan la presencia de algo frente a ellos.

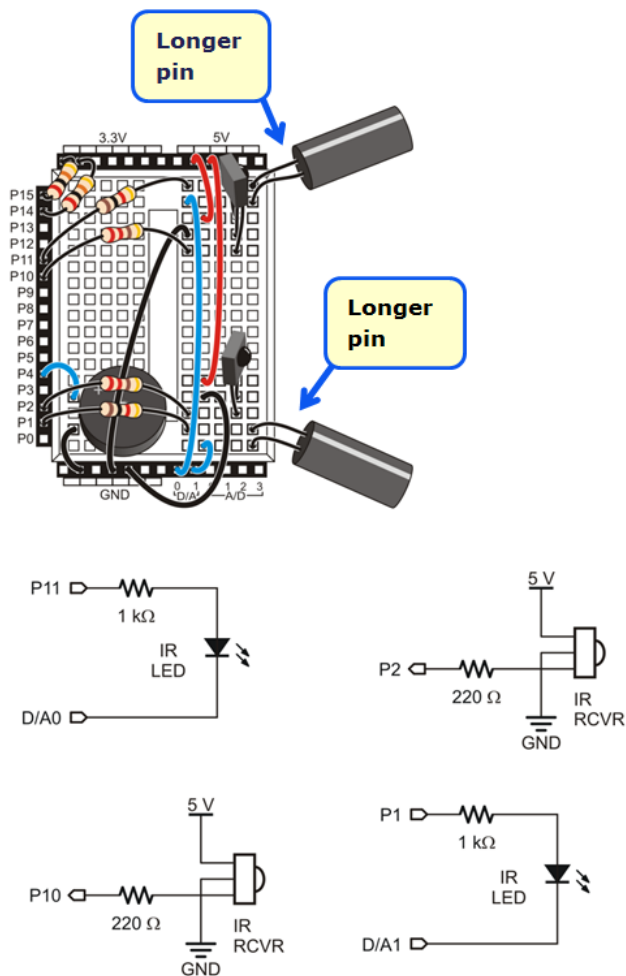
### Ultrasónicos



Se conectó en el puerto 8 con una resistencia de 2.22 Kohm la entrada de señal del sensor. Este fue utilizado para evitar que el robot golpeará algo al frente. Para información mas detallada puede ingresar a:

<http://learn.parallax.com/tutorials/robot/activitybot/activitybot/navigate-ultrasound>

## Infrarrojos



Siguiendo el esquema mostrado anteriormente se ensamblaron y colocaron los infrarrojos al robot, con la variación de tanto sus receptores como los emisores estaba cada uno hacia un lado del robot, esto permitió que no fuese necesario tener que girar en cada una de las comparaciones para confirmar si los lados estaban libres o no. Para información detallada del uso y ensamblaje de los infrarrojos, puede visitar:

<http://learn.parallax.com/tutorials/robot/activitybot/activitybot/navigate-infrared-flashlights>

Ambos dispositivos cuentan con ejemplos para probar su buen funcionamiento. Para mayor información puede visitarse [www.parallax.com](http://www.parallax.com)

## **La descripción del desempeño de la estructura de datos y el algoritmo utilizado cuando fue implementado en el robot comparado con la simulación hecha en la fase 1 del proyecto.**

El primer algoritmo fue hecho en el programa RUR-PLE, el cual tiene un programa basado en beepers los cuales fueron colocados con anterioridad en cada uno de los cruces que existían en el laberinto. El beeper puesto en la intersección constaba de un número #3 multiplicado por las salidas posibles que podía tomar el robot. Al final, el desempeño fue ejemplar únicamente debido a que se le dieron las opciones por donde podía ir el robot de RUR-PLE. Si el robot no hubiera tenido beepers para salir del laberinto, hubiera sido imposible resolverlo.

El segundo algoritmo, el cual se realizó en SimpleIDE tenía un desempeño bueno en laberintos de espacio estrechos. Es decir, con el laberinto tomado en la clase no tuvo el mejor desempeño debido a que existían lugares abiertos. En estos lugares, el robot por cada espacio vacío que encontraba, lo tomaba como un camino posible que podía tomar. Sin embargo el robot podía tomar las pequeñas plazas como posibles caminos infinitos, o también regresaba y se estrellaba contra la pared por los espacios existentes entre las paredes, los cuales los tomaba también como posibles caminos.

El Stack funciona genial pero en condiciones óptimas para el robot, debería de ser un lugar totalmente cerrado y sellado para que no existan posibilidades de confundir un pequeño espacio entre las paredes como un posible camino. Los stack funcionan, entonces, para regresar, en dado caso encuentre un callejón sin salida. Este regresara a la intersección más cercana tomada y tomara el camino que no había tomado.

### **Link del video saliendo de un laberinto.**

<https://www.youtube.com/watch?v=0JKpZEGfkiQ&feature=youtu.be>

## **Referencias:**

Imágenes e información obtenida en:

Parallax, Inc. (s.f.). <http://media.digikey.com/>. Recuperado el 28 de agosto de 2015, de

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

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