

Proyecto #1  
Segunda Fase



**Grupo 1**

Erick Bautista 15192  
Gerardo Molina 14492  
Diego López 141222

Universidad del Valle de Guatemala  
Guatemala Lunes 5 de Septiembre de 2016

# CONTENIDO

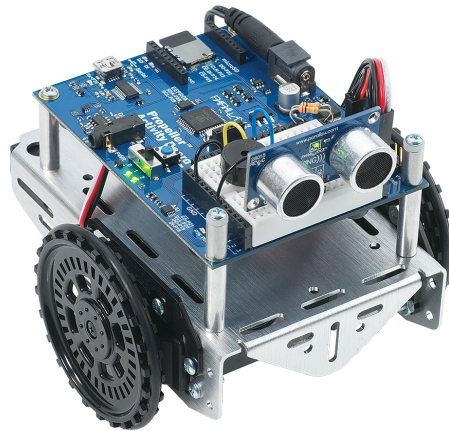
## CONTENIDO

- [1. Información del robot](#)
- [2. SimpleIDE](#)
- [3. Algoritmos Para Salir de un Laberinto](#)
- [4. Algoritmo de la Mano Derecha](#)
- [5. Componentes utilizados](#)
- [6. Funciones Utilizadas](#)
- [7. Pseudocódigo](#)
- [8. Desempeño de la Estructura](#)
- [9. Recomendaciones](#)
- [10. Enlace al Repositorio](#)
- [11. Referencias Bibliográficas](#)

## 1. Información del robot

El robot “Activity Bot” de Parallax Inc.<sup>1</sup> **Figura 1** utilizado para este proyecto es un hardware compacto de fácil implementación que puede ser programado con el lenguaje C. Este es utilizado para el aprendizaje, principalmente en proyectos que buscan poner a prueba su versatilidad y los conocimientos de programación y electrónica del estudiante.

El “Activity Bot” debe ser ensamblado, calibrado y programado previamente antes de poder ser utilizado, para en la caja del robot se incluye un manual el cual indica todos estos pasos. Si su robot no incluye o usted no cuenta con dicho manual puede ingresar al siguiente enlace [Manual Activity Bot](#).



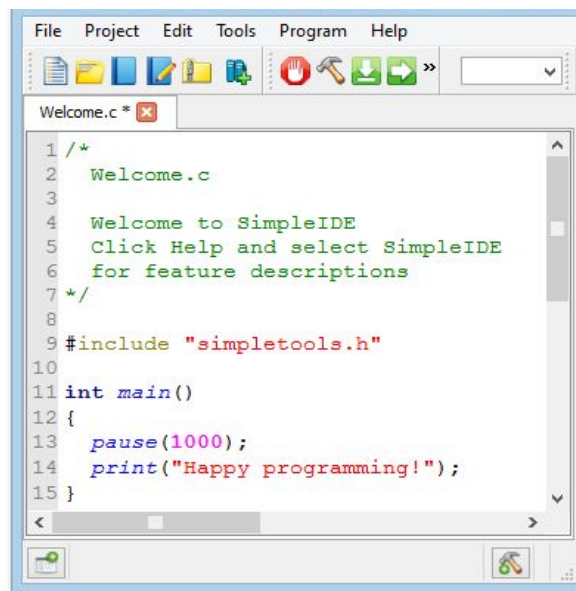
**Figura 1:** Activity Bot de Parallax inc.

## 2. SimpleIDE

La interfaz utilizada para la programación “Activity Bot” es la SimpleIDE **Figura 2** un ambiente de programación en C para este microcontrolador. Esta incluye todas las librerías y ejemplos para calibrar y entender el entorno del robot. Esta puede ser descargada en el siguiente enlace [SimpleIDE](#). También se pueden seguir algunos tutoriales en el siguiente enlace [Tutorial](#).

---

<sup>1</sup> Parallax Inc. es una compañía en California que fabrica y venden microcontroladores de varias gamas. Este es un enlace a su sitio web oficial: <https://www.parallax.com/>



**Figura 2:** captura de SimpleIde.

### 3. Algoritmos Para Salir de un Laberinto

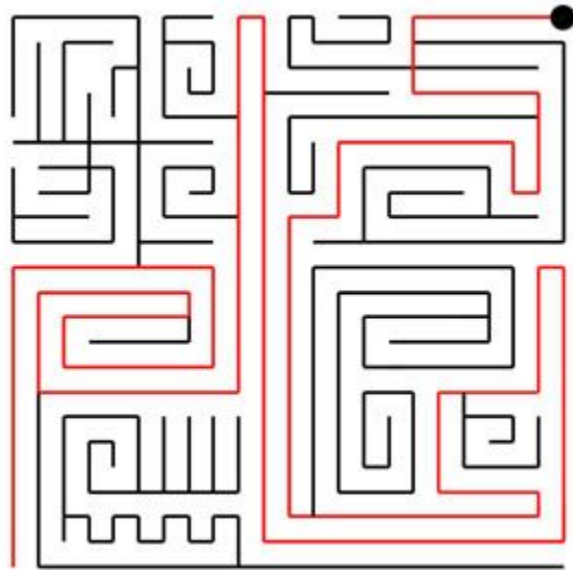
Para que el robot logre salir de un laberinto se investigó varios tipos de algoritmos ya existentes entre ellos:

- Algoritmo Backtracking
- Algoritmo Tremaux
- Algoritmo de la Mano Derecha

Luego de haber investigado estos y algunos otros más se tomó la decisión de trabajar con el Backtracking, también conocido como vuelta atrás pero luego de analizar bien las posibilidades de que el robot no fuera capaz de girar completamente  $180^\circ$ , al tener solamente un sensor iba a ser muy probable que el robot chocara y se quede atascado con una pared. Luego se consiguieron otros dos sensores por lo cual se siguió con la idea de este mismo algoritmo, luego de varios intentos y fracasos se llegó a la conclusión que utilizar el backtracking era una opción poco eficiente en cuanto al tiempo del recorrido por lo que sería muy probable que el robot no saliera en los 5 minutos otorgados. En base a esto se tomó la decisión de cambiar al algoritmo de la mano derecha.

### 4. Algoritmo de la Mano Derecha

El algoritmo de la mano derecha consiste en seguir siempre la pared derecha (también funciona con la izquierda) hasta llegar al final del laberinto. Este método es uno de los más antiguos conocidos para resolver laberintos, o caminos que así lo parezcan. Esta resolución depende mucho del punto inicial dentro del laberinto por lo que el tiempo que se tarda en ser solucionado puede variar.



**Figura 3:** diagrama de resolución por medio de la mano derecha.. <sup>2</sup>

## 5. Componentes utilizados

- **Motores Servo:** el robot implementa dos motores servos hechos específicamente para este robot, por lo que son de gran velocidad y se especializan en recibir la información de una manera eficiente para tomar varias posiciones rápidamente.
- **Sensores Ultrasónicos:** se utilizaron tres sensores ultrasónicos, uno de ellos es el sensor parallax ultrasónico 28015 **Figura 4** y también se utilizaron dos sensores *hcsr04* **Figura 5** a los cuales se les realizó una adaptación en paralelo con una resistencia de 10k para que funcionara de igual manera que el anterior. Se puede visitar el datasheet de este en el siguiente enlace [hcsr04](#).
- **Resistencias:** se utilizaron resistencias de 10k y 2.2k ohms para conectar los sensores al robot y realizar la adaptación de los sensores.

---

<sup>2</sup> Enlace de la imagen: <http://www.curiosidadsq.com/2012/08/como-salir-de-laberintos.html>



**Figura 4:** sensor ultrasónico Parallax 28015.



**Figura 5:** sensor ultrasónico hcsr04

## 6. Funciones Utilizadas

(Descripción de las funciones que usamos en simpleIDE)

Centro():

- La función centro permite que el robot se centralice respecto a su posición.
- Se realizó dicha función para que el robot siempre se centre y no colisione con las paredes a su alrededor.

GDerecha():

- Esta función hace girar al robot 90 grados a la derecha.

GIzquierda()

- Esta función hace girar al robot 90 grados a la izquierda.

Adelante()

- Si el robot detecta pared a los lados sigue recto.

Adelante2()

- Esta función fue creada para que el robot no choque con la pared al girar cerca de una pared.

Paro()

- Esta función para cualquier movimiento del robot.

Paso()

- Esta función le dice al robot si debe girar a la derecha, izquierda o seguir adelante.

PreAnalizar()

-

Analizar()

## 7. Pseudocódigo

(Pseudocodigo)

## 8. Desempeño de la Estructura

(Desempeno comparado con el desempeno de la primera fase)

## 9. Recomendaciones

(recomendaciones a personas que utilizan el robot en el futuro o harán el proyecto)

## 10. Enlace al Repositorio

Click aquí para visitar el enlace a nuestro proyecto--> [Enlace al Repositorio Github](#)

## 11. Referencias Bibliográficas

- Parallax, Inc. s.f. *Activity Bot*. 2 de Septiembre 2016. [www.media.digikey.com](http://www.media.digikey.com)

- Parallax, Inc. s.f. *Activity Bot*. 2 de Septiembre 2016.  
<https://media.digikey.com/pdf/Data%20Sheets/Parallax%20PDFs/ActivityBot.pdf>
- Parallax, Inc. s.f. *Página Principal*. 2 de Septiembre 2016. <https://www.parallax.com/>
- S.A. 2015. *Backtracking*. 2 de Septiembre 2016. [https://en.wikipedia.org/wiki/DPLL\\_algorithm](https://en.wikipedia.org/wiki/DPLL_algorithm)
- Parallax, Inc. 2013. *Set Up SimpleIDE*. 2 de Septiembre 2016.  
<http://learn.parallax.com/tutorials/language/propeller-c/propeller-c-set-simpleide>
- ElecFreaks.s.f. *HC-SR04*. 2 de Septiembre 2016.  
[www.micropik.com/PDF/HCSR04.pdf](http://www.micropik.com/PDF/HCSR04.pdf)