

Instructivo para salir de un laberinto con Parallax Activitybot



Universidad del Valle de Guatemala

María Fernanda López Díaz

David Uriel Soto Alvarez

Hector Javier Carpio Garcia

Guatemala, Marzo 2018

Tabla de contenido

1. Arranque rápido

- a. Instalación de software
- b. Configuración del entorno del Activitybot
- c. El lenguaje C
- d. Importación de librerías

2. Saliendo del laberinto

- a. Sensores
- b. Servos y/o motores
- c. Parámetros para salir del laberinto

3. Implementación

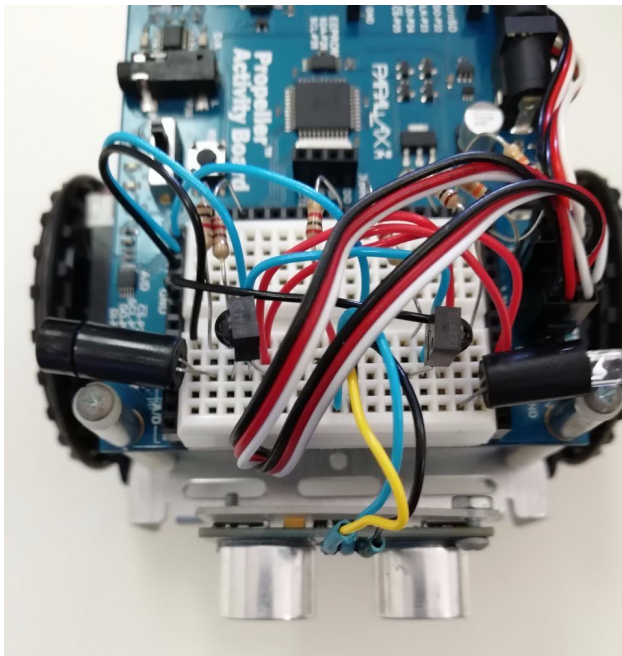
- a. Desempeño estructura de datos
- b. Descripción algoritmo utilizado

Arranque rápido

Instalación de software

El procedimiento exacto para instalar el IDE depende del sistema operativo que esté utilizando, pero el principio básico es el mismo. Instale el IDE que se acople a su sistema operativo junto con los USB drivers y reiniciar el IDE. Puede descargar la versión del software que necesite en el siguiente enlace: <https://www.parallax.com/downloads/simpleide-software-windows-propeller-c>. Para descargar los drivers visite el siguiente enlace: www.parallax.com/usbdrivers

Cableado



Lenguaje C

En el entorno del ActivityBot sus instrucciones se programan con el lenguaje C. Este lenguaje es igual de tipado que Java por lo que la sintaxis es similar. Para conocer más acerca del lenguaje C y como implementarlo en el IDE y el robot visite el siguiente enlace: <https://learn.parallax.com/tutorials/language>.

Importación de librerías

Las librerías a utilizar son la de "Simpletools.h", "abdrive.h" y "ping.h". Las primeras dos mencionadas sirven para obtener las herramientas y funciones básicas del Parallax, como los pines del protoboard que ya están integrados, y funciones como `drive_goto()` y `drive_speed()`. Estas primeras librerías ya están predeterminadas al iniciar un nuevo proyecto.

La librería de ping.h nos permite utilizar los sensores incorporados. Esta librería es capaz de llamar el sensor ubicado en frente del robot para determinar distancias entre el Parallax y lo que se encuentra en frente de él, y el infrarrojo de la derecha para que este pueda cruzar cuando se le indique.

Saliendo del laberinto

Sensores

El sensor que se utilizará para lograr que el robot salga de cualquier laberinto es el sensor de distancia ultrasónico. Este sensor determina qué tan lejos está un objeto del robot, desde una distancia de tres centímetros hasta tres metros. En el caso del laberinto nos ayudará a encontrar donde se encuentra una pared para que así el robot se vaya guiando y cambiando de posición cuando no encuentre una pared. Para probar y aprender más acerca del sensor ultrasonico visite la página <https://learn.parallax.com/tutorials/language/propeller-c/propeller-c-simple-devices/sense-distance-ping>.

Parámetros y métodos para salir del laberinto

El algoritmo que se implementó para que el robot saliera de cualquier laberinto fue el de la mano derecha. Después de realizar las funciones necesarias se realizó un ciclo while para cuando el robot esté dentro del laberinto. Los parámetros que se utilizaron en este ciclo son:

1. Freqout: hace que la luz infrarroja de la izquierda se active con una potencia de 19000. Para más información de este parametro consultar el siguiente enlace:

<http://www.parallax.com/go/PBASICHelp/Content/LanguageTopics/Commands/FREQOUT.htm>

En el método principal se activan los puertos de digital a análogo en bajo, con low(26) y low(27).

Implementación

Estructura de datos

La estructura de datos que se utilizó en la implementación del algoritmo fue la utilización de métodos, entre los cuales están:

- `move()`: en esta se encuentra `drive_goto()` en la cual permite el movimiento de las ruedas.
- `turn_left()`: este método hace avanzar la rueda derecha, y hace retroceder la izquierda para permitir el giro.
- `turn_right()`: este método hace avanzar la rueda izquierda, y hace retroceder la derecha para permitir el giro.
- `find_path()`: si el Parallax encuentra un objeto enfrente y uno a la derecha, gira a la izquierda.
- `follow_right_wall()`: Si no encuentra nada a la derecha, gira a la derecha.

Descripción del algoritmo

El algoritmo consiste en seguir la pared derecha, y avanza hasta encontrar un espacio a la derecha y gira a la derecha hasta encontrar una pared nuevamente en la derecha. Luego, si encuentra una pared enfrente gira a la derecha primeramente. Finalmente, si encuentra una pared enfrente y una a la derecha, gira a la izquierda.