

# ActivityBot

## Instructivo

### Instalar SimpleIDE Software

Es un programa open source que permite programar en C. Está diseñado para los principiantes.

1. Seleccionar el sistema operativo

#### Click on your system to get started

- [Windows](#)
- [Mac](#)
- [Linux](#)
- [Raspberry Pi](#)

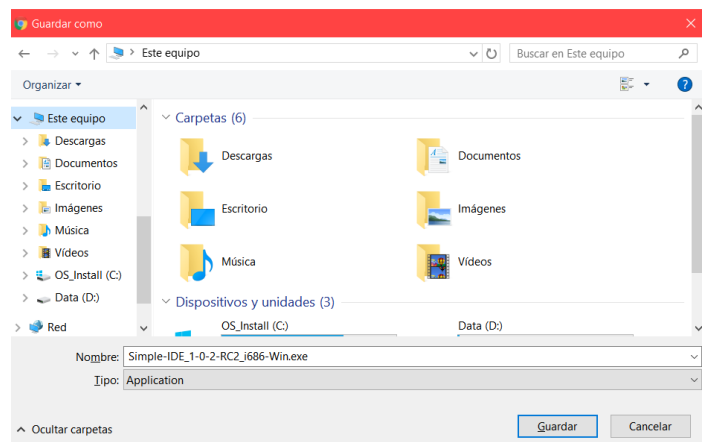
2. Al seleccionar Windows como sistema operativo, nos envía a esta pestaña

#### Windows

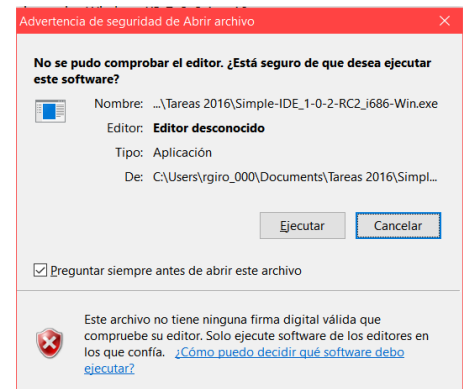
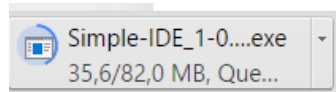
##### Windows Installation

- ✓ Verify that your computer is running Windows XP, 7, 8, 8.1 or 10.
- ✓ If you have Windows 10, download the Windows 2.12.14 (or higher if available) driver from FTDI's [VCP Drivers](#) page, and install it first.
- ✓ [Download SimpleIDE 1.0 RC2 for Windows](#)
- ✓ Run the installer. For best results with the tutorials, accept all default settings.

3. Dar Click en el recuadro de "Download SimpleIDE 1.0 RC2 for Windows" y preguntará dónde guardarlo



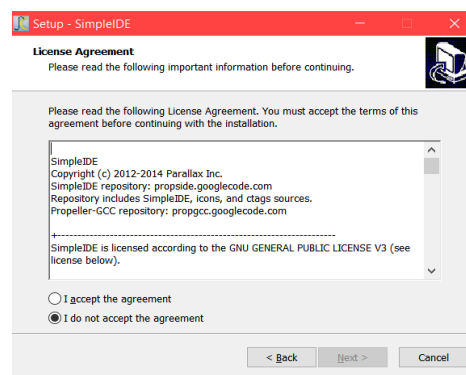
4. Cuando se descargué, darle Click a esté recuadro y aparecerá una advertencia. Debe darle click a “Ejecutar”



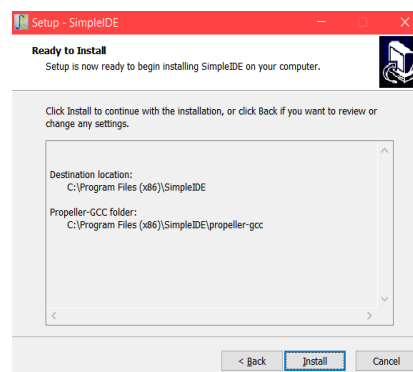
5. Dar Click en “Next”



6. Aceptar los términos y darle Click en “Next”



7. Darle click en “Install”



8. Y por último, aparecerá el SimpleIDE para empezar a programar

```

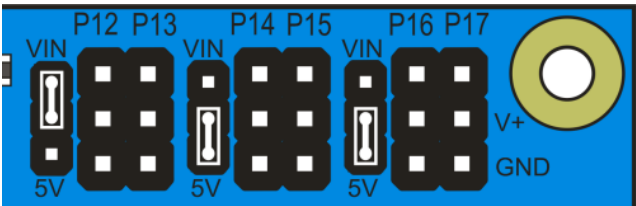
1 /*
2  Welcome.c
3
4  Welcome to SimpleIDE, the C programming environment that makes it easy to
5  get started with the multi-core Propeller microcontroller!
6
7  To run this program:
8
9  - Is this the first Parallax software you've installed on your computer?
10 If so, install USB driver's now: www.parallax.com/usbdri
11 - Connect your Propeller board to your computer's USB. Also connect power
12 to the board if needed, and if it has a power switch, turn it on.
13 - Make sure to select your COM port from the toolbar dropdown menu in the
14 upper-right corner. If you are unsure which COM port your board is
15 connected to, disconnect it and click the dropdown to check the port
16 list, then reconnect it and check again.
17 - Click Program and select Run with Terminal (or click the Run with Terminal
18 button). The SimpleIDE Terminal should appear and display the "Hello!"
19 message.
20
21 Next, check the Help menu for new online tutorials, software manual, and
22 reference material.
23 http://learn.parallax.com/propeller-c-tutorials
24 */
25
26 #include "simpletools.h"           // Include simple tools
27
28 int main()                       // Main function
29 {
30     print("Hello!");             // Display test message
31 }

```

# Preparación del robot

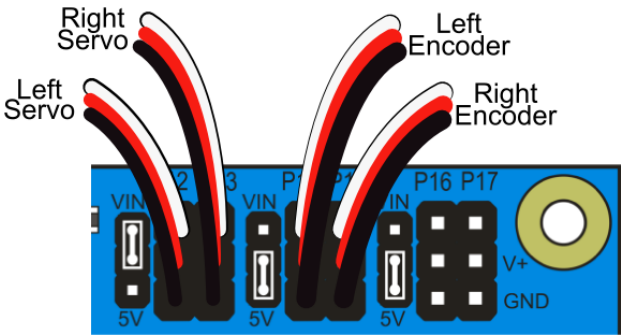
## Preparar los puertos de los servos

- P12 & P13: VIN
- P14 & P15: 5V
- P16 & P17: 5V



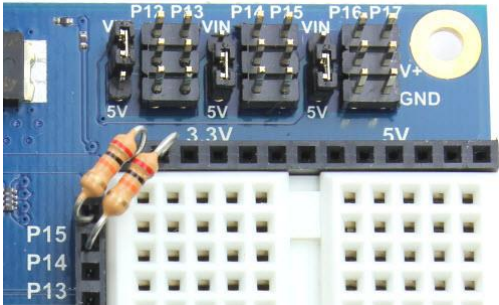
## Conectar los cables

- P12: Servo Izquierdo
- P13: Servo Derecho
- P14: Encoder Izquierdo
- P15: Encoder Derecho



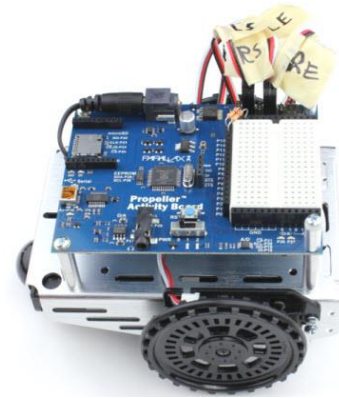
## Agregar Resistencias

- Usando la resistencia de 20 k-ohm, conéctelo a P14 y P15, a la izquierda del protoboard hacia 3.3 V



Inserte las baterías

- Coloque 5 baterías AA de 1.5 V en la cajuela de las baterías



## Información de Sensores

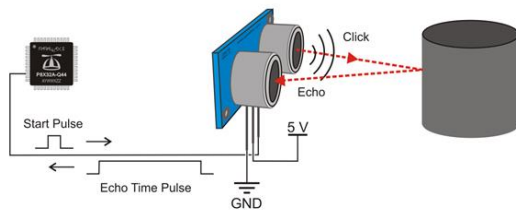
The PING))) sensor



Este sensor solo necesita corriente y tierra; y una señal para obtener la distancia respectiva.

Proporciona un método sencillo de medición de distancia. Este sensor es perfecto para cualquier número de aplicaciones que requieren que se realice mediciones entre objetos en movimiento o estacionarios.

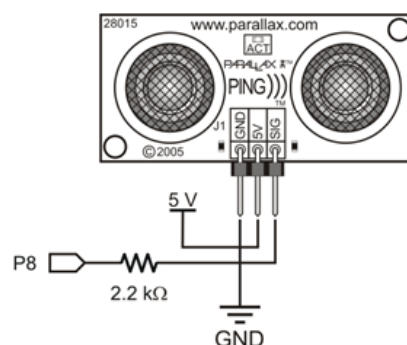
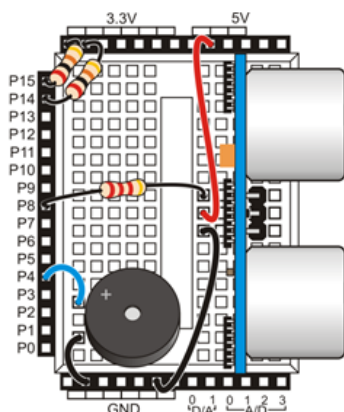
Funciona haciendo un ruido de alta frecuencia, la medición se basa en el tiempo que se necesita para escuchar el eco de retorno. Por lo tanto, las medidas dependen de la velocidad del sonido en el aire.



La velocidad del sonido cambia con la temperatura del aire, humedad, e incluso la calidad del aire. Ni la calidad ni la humedad del aire crea una diferencia suficiente como para figurar en el sensor PING))). La temperatura del aire, por otra parte, puede causar errores de distancia medibles.

La velocidad del sonido aumenta en 0,6 metros por segundo (m / s) para todos los (C °) aumento grados centígrados de temperatura.

Construir el PING))) Sensor



Asegurarse de usar la resistencia de 2.2 K ohms

## *Comparación con fase 1 y fase 2*

### **Backtracking**

El algoritmo de backtracking es utilizado para resolver problemas en los que la solución consta de una serie de decisiones adecuadas hasta un objetivo. De esta forma genera todas las secuenciadas de forma sistemática y organizada hasta encontrar la correcta.

### **Algoritmo de la mano derecha**

Este método consiste en simplemente mantener cualquiera de las dos manos pegada a la pared y avanzar hasta llegar a la salida. Este algoritmo funciona siempre que el laberinto tenga una conexión simple, lo que significa que todas las paredes están conectadas, y que exista una salida.

Se hizo una adaptación y cuando el robot de 5 giros hacía la derecha, seguirá recto hasta encontrar una pared y posteriormente girará 2 veces a la derecha ( $180^\circ$ ); para encontrar una salida.

### **Comparación**

Implementamos otro algoritmo porque el backtracking nos presentó demasiados problemas con las validaciones que se deben hacer.

El backtracking que usamos en la fase 1, verificaba en las paredes y por lo tanto se debían hacer demasiadas modificaciones.