Laboratorio de Robótica

Ricardo Illa

Universidad Católica del Uruguay

01/03/2016

Contenido

1. Resumen

El programa completo consiste en un sistema de control para robots móviles, realimentado mediante una cámara web, y que utiliza Wifi para su comunicación. Cada robot solo recibe los valores correspondientes a la velocidad de sus motores y el encargado de calcular dichos valores es un programa (servidor) que se ejecuta en una PC. El servidor además utiliza la imagen captada por la webcam para obtener la posición de cada robot en el plano.

Si bien el software ya viene con un control de posición integrado que deja elegir el SetPoint del robot, se prevé que el código se pueda editar y así comprobar en la práctica las diferentes estrategias de control estudiadas en el curso de Robótica. Es por eso que se plantea un manual de instalación de los entornos de desarrollo y sus bibliotecas necesarias.

2. Instalación de Lejos en EV3

Lejos es un sistema operativo realizado para poder programar cualquier Lego Mindstorm EV3 utilizando Java. El SO se instala en una tarjeta SD por lo que no es necesario nunca borrar el sistema operativo original de Lego.

Todo el procedimiento que se describirá a continuación puede verse claramente en un video.

Link: https://www.youtube.com/watch?v=NyoF0Ws6SkY

2.1. Materiales

- Memoria microSD de 2Gb mínimo

- JDK 1.7

- Lejos EV3 0.8.1, link: https://sourceforge.net/projects/ev3.lejos.p/files/

- JRE para Mindstorms link: http://www.oracle.com/technetwork/java/embedded/downloads/javase/javaseembeddedev3-1982511.html

2.2. Procedimiento

Primero que nada luego de haber instalado el JDK y haber descargado los demás archivos instalar el software Lejos EV3. Una vez finalizada la instalación nos saldrá un cuadro de dialogo. Desde este cuadro debemos crear la tarjeta SD booteable. Para eso necesitamos que nuestra tarjeta SD esté formateada en sistema Fat32, seleccionar el JRE desde la carpeta que lo guardamos y le damos crear.

Luego de haber creado la tarjeta, insertarla en el EV3 y encenderlo. En aproximadamente 10 minutos el sistema operativo debería quedar listo. Cuando termine se podrá ver el logo de LejosEV3 en pantalla.

3. Instalación de Eclipse y Plug-in Lejos EV3

3.1. Materiales

- Eclipse. Link: https://eclipse.org/downloads/

3.2. Procedimiento

Para comenzar a programar el EV3 es necesario instalar eclipse en un PC. Luego de tenerlo instalado, entrar en Help/add new software, click en add e introducir: http://lejos.sourceforge.net/tools/eclipse/plugin/ev3 finalmente seleccionarlo e instalarlo y dejar que el programa se reinicie.

Una vez que eclipse se vuelve a abrir debemos ir a al menú Window, luego preferences, y en donde dice EV3_Home debe estar la ruta hacia la carpeta donde está instalado Lejos Ev3.

3.3. Crear y Cargar una aplicación

Para poder cargar una aplicación en el Ev3 es necesario un dongle Wifi compatible como por ejemplo el NETGEAR WNA1100. Una vez que se tiene colocado el dongle en el robot hay que conectarse a una red inalámbrica (Si se utiliza el router de siempre, el SSID es *Lego* y la contraseña es *aaaaaaaa*). Para conectarse ir al menú de

Lejos en la parte de Wifi, escribir la contraseña de la red y luego clickear sobre el ícono de Siguiente. Una vez conectado aparecerá en su pantalla la IP proporcionada por el DHCP.

Posteriormente conectar al PC en la misma red que está el EV3, y en el menú de eclipse entrar a Windows/Preferences/EV3 para completar el campo Name con la IP del Robot.

Ahora se debe crear un nuevo proyecto de LeJOS en Eclipse, para ello ir a File, New Project y seleccionar LeJOS EV3 Project. Luego dentro del proyecto ya se pueden crear las clases. Un ejemplo sencillo se muestra en la figura 1.

```
import lejos.hardware.BrickFinder;
import lejos.hardware.lcd.GraphicsLCD;
import lejos.utility.Delay;

public class FirstApp {

    /**
     * @param args
     */
    public static void main(String[] args) {
        GraphicsLCD g = BrickFinder.getDefault().getGraphicsLCD();

        g.drawString("Hello World", 0, 0, GraphicsLCD.VCENTER |
                GraphicsLCD.LEFT);

        Delay.msDelay(5000);
    }

}
```

Figura 1. Código ejemplo en java para EV3.

Finalmente para correr el programa se debe hacer click derecho sobre el código a ejecutar, y seleccionar Run as/LeJOS EV3 program.

4. Netbeans

Netbeans es el entorno de desarrollo donde se creó el programa principal. Si bien se puede ejecutar el programa principal también desde eclipse, se recomienda instalar Netbeans por su simplicidad.

Instalarlo es sencillo y se puede descargar desde la página web del autor: https://netbeans.org/

Para que el programa funcione debemos además incluir las librerías Jama 1.0.3.jar, y OpenCV 2.4.11.jar disponibles en: http://math.nist.gov/javanumerics/jama/ y http://opencv.org/downloads.html respectivamente.

Para incluir las librerías se debe tener el proyecto abierto, luego hacer click derecho en Libraries y finalmente click en Add jar/Folder, tal como se muestra en la figura xx. Finalmente el proyecto se verá como en la figura xx.

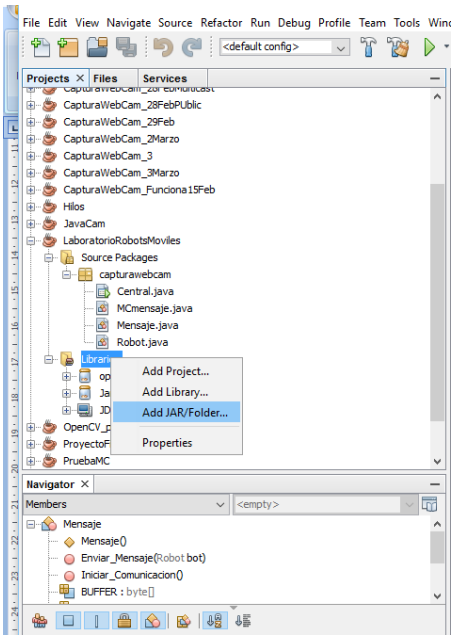

Figura 2.1 Agregado de librerías jar.

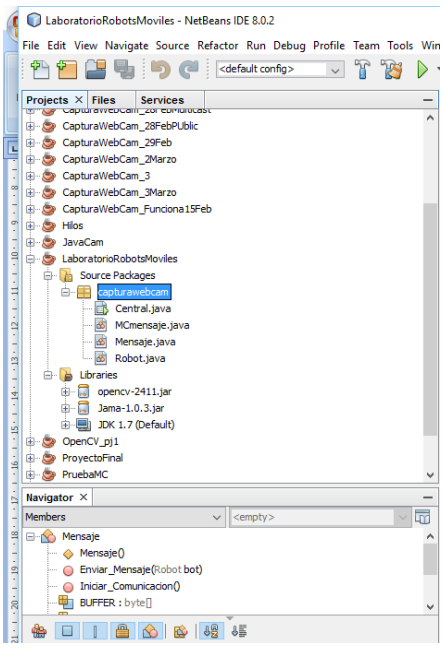

Figura 2.2 Proyecto completo.

5. Descripción del programa completo

5.1. Software para Lego EV3

El software creado para cada robot está organizado dentro de una clase llamada Main que se encarga de crear cada hilo y comenzar su ejecución. El resto de las clases heredan de la clase Thread, lo que quiere decir que funcionan en “paralelo”. En total son 3 hilos: Comunicación, Comando y Exit.

Comunicación como su nombre lo dice se encarga de recibir la información por wifi, utilizando UDP. En este caso el robot solo recibe mensajes continuamente y los decodifica.

Comando es el hilo encargado de dirigir los motores del robot, una vez que decodifica la información, setea las velocidades de cada motor. La librería de LejOS ya contiene un método llamado Motor.setSpeed(int) que se encarga de todo el resto del proceso.

Por último el hilo llamado Exit es necesario por si en algún momento se desea interrumpir la ejecución del programa. Simplemente presionando el botón Escape del robot, finaliza el proceso.

5.2. Software para servidor (PC)

5.2.1. Modo de uso

Primero que nada el programa principal posee interfaz grafica, ya que es con lo que el usuario interactúa. Dentro de la interfaz grafica se puede encontrar el panel donde aparece la imagen de la cámara web, además de botones y campos de texto.

Las llamadas variables fijas son:

- **Umbral**, debe ser un número entre 0 y 255, define el umbral de binarización de la imagen, es decir que tan oscura quedará. Puede cambiarse en cualquier momento de la ejecución.
- **Número de robots**, esta variable debe ser fijada al comienzo de la ejecución para que no se produzcan errores.
- **Velocidad máxima,** es el tope de velocidad a la que pueden circular los robots, generalmente entre 0 y 300. Puede cambiarse en cualquier momento de la ejecución.

También aparecen 4 botones:

- **Actualizar variables**, cuando cualquiera de las variables es modificada, luego se debe hacer click en este botón para que se actualicen.
- **Auto-Calibración**, Antes de iniciar el programa se puede calibrar el factor de conversión entre pixeles y metros con este botón. Simplemente debe haber un robot frente a la cámara y el programa guía al usuario para la calibración. El programa utiliza la distancia conocida entre los puntos blancos del robot.
- **SetPoint,** botón que permite cambiar el punto destino del robot. Para utilizarlo se debe hacer click en cualquier lugar del panel de video, y luego hacer click en este botón.
- **Inicio,** es el botón para encender el control.

Por último el panel en blanco es un monitor donde el programa despliega mensajes.

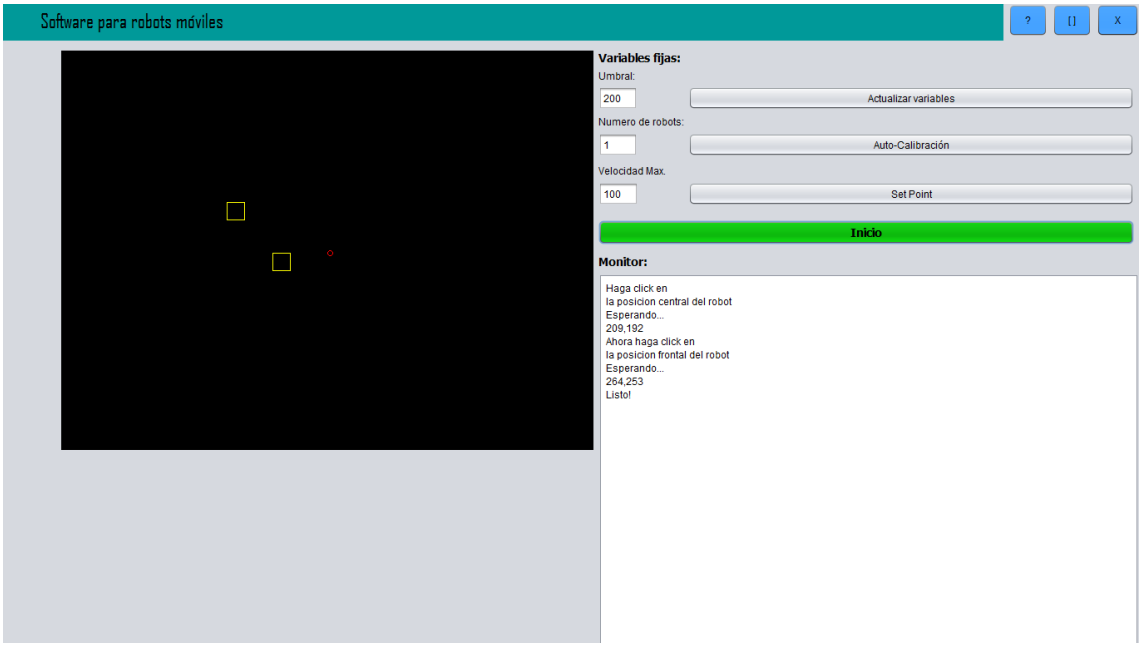

Figura 3. Interfaz gráfica del programa principal.

5.2.2. Funcionamiento

Al igual que en el software instalado en el EV3 este programa también contiene clases tipo Thread. Por un lado está la interfaz gráfica, y por otro lado está el hilo principal (llamado hilo1). El hilo1 utiliza clases como la comunicación llamada MCmensaje por MultiCast, y la clase Robot. A su vez cada botón tiene un evento asociado que llama a los métodos correspondientes.

El programa en general comienza creando un array de objetos tipo Robot, luego pide al usuario que indique la posición inicial de cada robot haciendo click en cada uno de sus puntos. Posteriormente comienza el reconocimiento, es decir la asociación de los robots con sus posiciones. Lo que hace es pedir que se mueva el robot con id=n, luego detecta el movimiento y lo asocia, así sucesivamente con n variando hasta la cantidad de robots indicados. Si el programa no contara con esta función no sabría nunca donde está el robot1, donde está el robot2, etc.

Una vez detectados y asociados los robots, se entra en un bucle infinito que obtiene la posición de cada robot en pixeles, transforma esa posición en una postura, (*x,y,thita*) y envía estos datos al método Control. El método control calcula las velocidades de cada robot y por último se envía el resultado a través de wifi.

Los mensajes son enviados por Multicast, es decir que la información le llega a todos los robots por igual. La información es codificada en una cadena de caracteres

que contiene las ids de cada robot, seguido de sus dos velocidades, por ejemplo: (“*id1,130,140,id2,120,115,id3,13,14, ...”)*
Para entender con mayor profundidad basta con explorar el código de java.