

Bogotá 22 de noviembre del 2011

**TEATRO DE LA MEMORIA EN MUNDOS VIRTUALES**  
**UNIVERSIDAD NACIONAL DE COLOMBIA**  
**SEDE BOGOTÁ**

**ENTREGABLE #1**

César Augusto Pantoja

Estudiante auxiliar ingeniería electrónica

1. Conexión y organización del cableado del circuito de control y de los motores del brazo robot:

El trabajo consiste en terminar las conexiones faltantes de los motores y de las tarjetas electrónicas de control, para lo cual se procedió de la siguiente manera:

El brazo se encontró completamente desconectado (figura 1), aunque algunos de los cables que comunican los motores con las tarjetas electrónicas de control ya se habían acondicionado dentro de una coraza plástica, aún faltaba terminar de conectar los encoder de los motores 3, 4 y las tarjetas electrónicas de control.



Figura 1.

Lo primero fué amarrar las corazas a la estructura, teniendo en cuenta dejar suficientes vueltas de cable para que el brazo pueda girar libremente sin enredarse. (Figura 2)



Figura 2.

Los motores se encuentran numerados tal como se muestra en la figura 3.

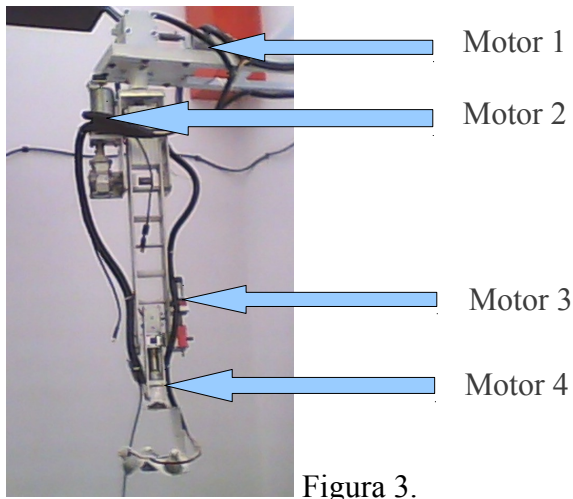


Figura 3.

Teniendo en cuenta esta numeración se procede a la conexión de los motores y los encoder. Los motores 1, 2 y sus respectivos encoder se encuentran ya listos y sólo era necesario conectarlos. En los motores 3, 4 fué necesario obtener conectores tanto para los motores como para los encoder, (figuras 4 y 5)

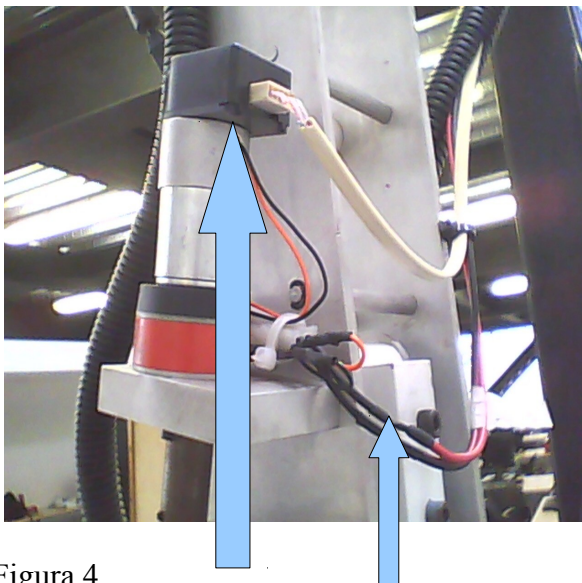


Figura 4  
Motor #3

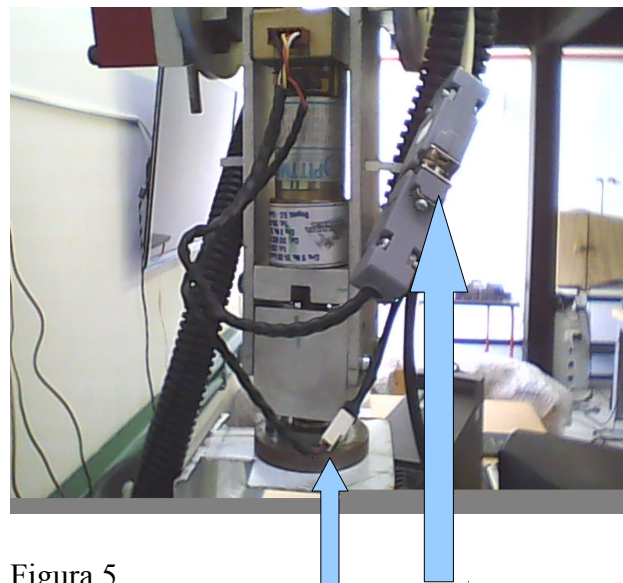
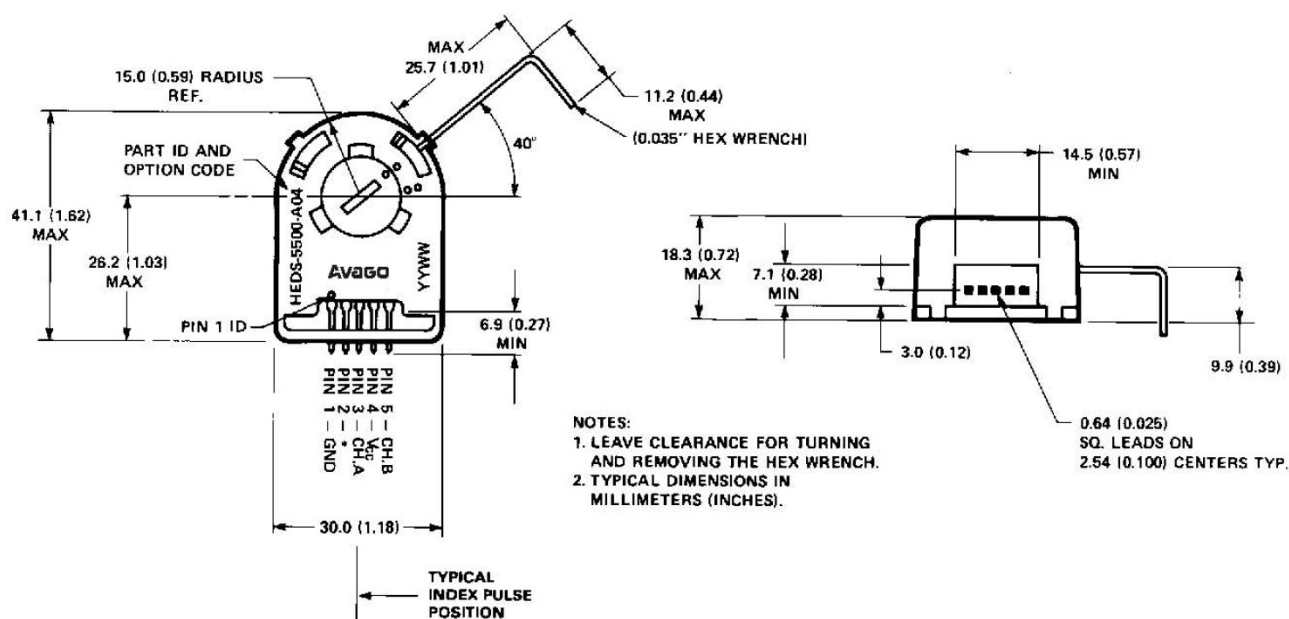


Figura 5  
Motor #4

Los encoder se conectaron teniendo en cuenta el siguiente pinout:  
Encoder motor #3 marca HEDM-55XX (mismo encoder de los motores 1 y 2), según el datasheet:

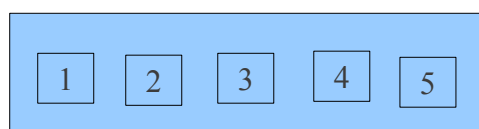




#### Encoder motor #4

Es un motor marca Pittman referencia GM8724S011 con encoder incorporado, el cual tiene el siguiente pinout:

NOTA: Este pinout es igual al del encoder HEDM.



1. GND
2. N/C
3. CHA
4. VCC
5. CHB



Entrada conector

Después de sujetar las corazas y de conectar todos los motores, se procede a la conexión de las tarjetas electrónicas de control:

NOTA: Esta conexión se basa en la tesis de grado de:

FERNEY ALEJANDRO ALVAREZ BENITEZ

Título: DISEÑO Y CONSTRUCCIÓN DE PARTES DE HARDWARE PARA EL PROYECTO: TEATRO DE LA MEMORIA EN MUNDOS VIRTUALES, 3a FASE.

Director: GERMAN ANDRÉS RAMOS

UNIVERSIDAD NACIONAL DE COLOMBIA, BOGOTÁ D.C. 2007

#### 2. Conexión de las tarjetas de control:

El tablero de control contiene hasta el momento (22 de noviembre del 2011) lo siguiente:

Cantidad	Descripción
1	Tarjeta PCB TMS320F2808 (DSP)
1	Tarjeta PCB interfaz con DSP
1	Tarjeta PCB con el circuito lector de encoders
2	Tarjetas PCB servoamplificadoras
1	Fuente voltaje conmutada de entrada 115VAC salida: 24VDC
1	Interrupor de 3 contactos en caja plástica

La distribución de las tarjetas electrónicas de control se muestran en la figura 6.

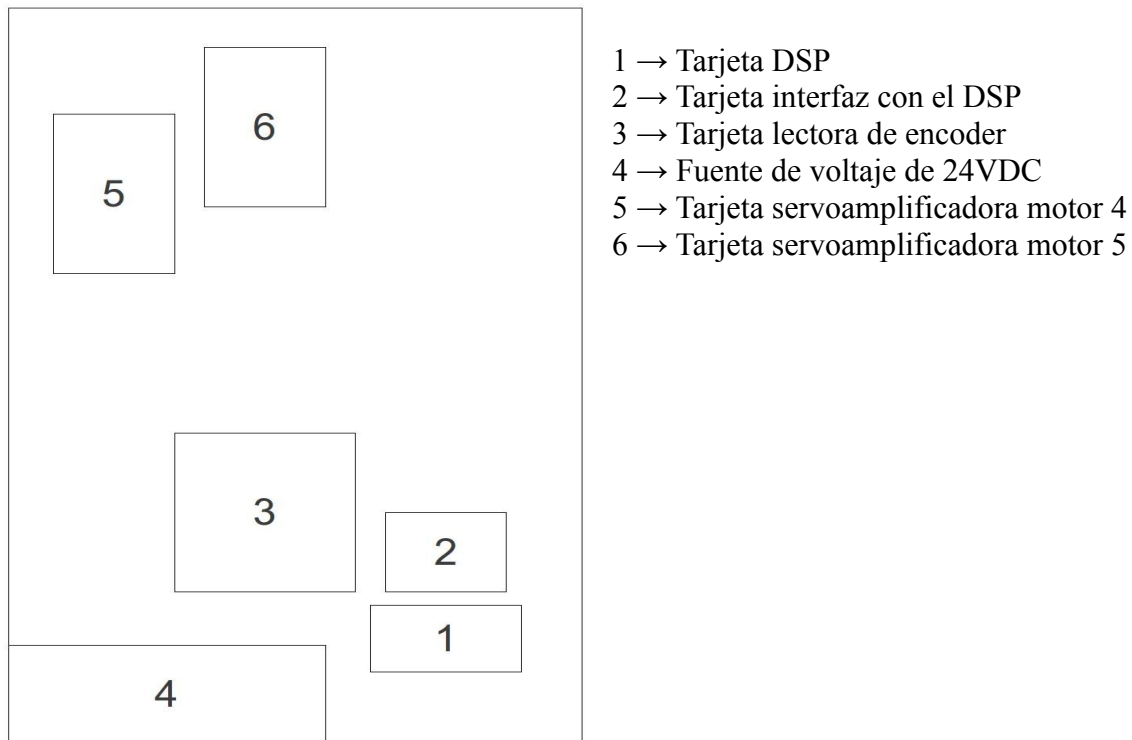


Figura 6.

La conexión entre las tarjetas 1 y 2 se realiza con base en el esquema de la figura 7.

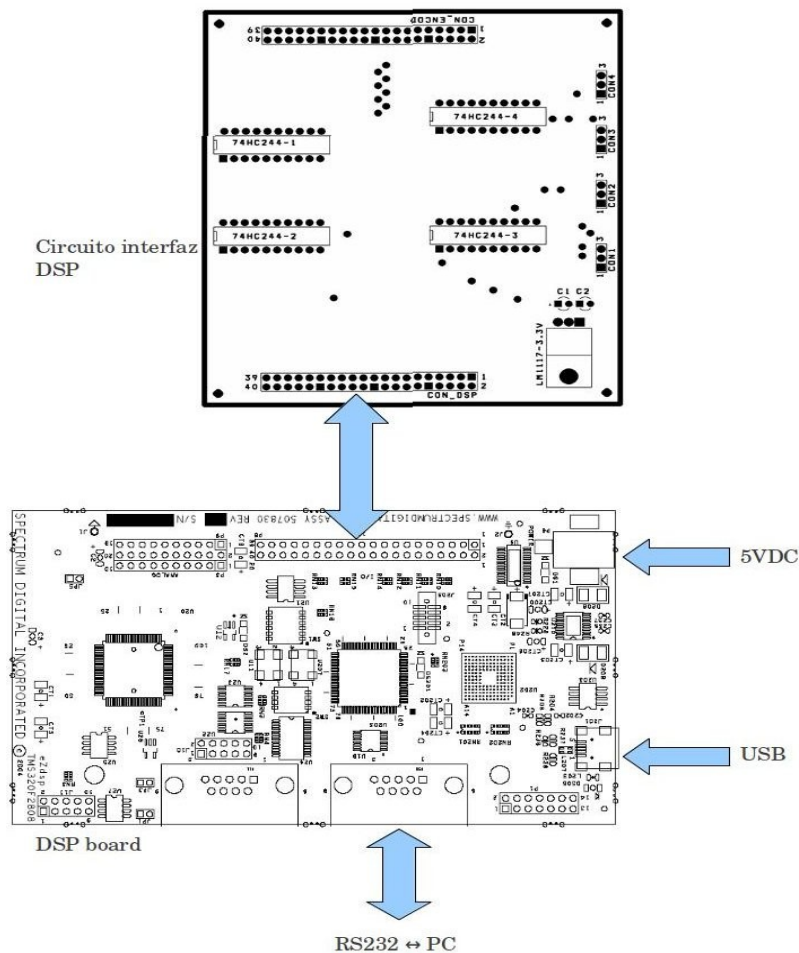
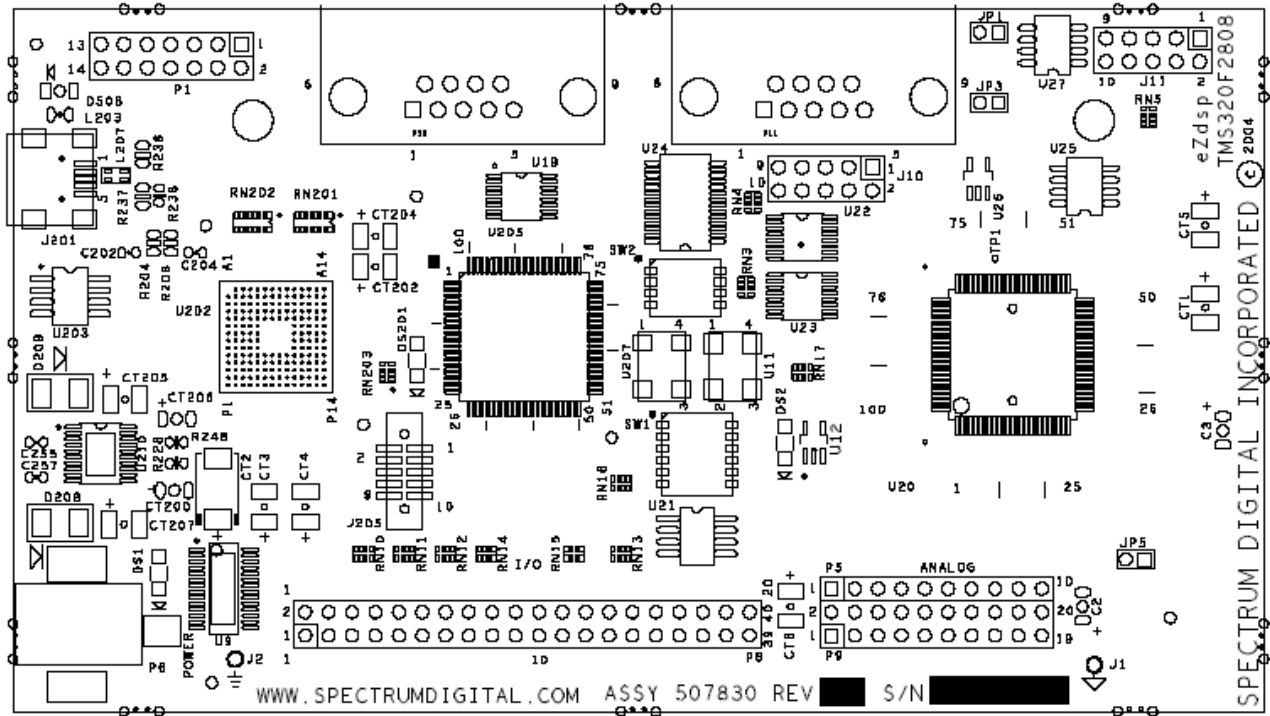


Figura 7.

#### Tarjeta DSP:

Los motores se controlan a través de un DSP, el cual recibe las instrucciones a través del puerto serial de una PC con una aplicación JAVA que interpreta los movimientos de un joystick y los convierte en órdenes para el movimiento del podio.

La tarjeta electrónica que contiene el DSP es de referencia TMS320F2808 de Digital Spectrum.



DSP board digital spectrum

Esta tarjeta contiene 2 puertos seriales nombrados SCIA y SCIB, para el control del podio se utilizó el puerto SCIA, al cual se le fabricó el cable con la siguiente conexión:

#### DB9 SCIA (DSP)

#### DB9 RS232 (PC)

pin	pin	info
2 -----	2	Rx
3 -----	3	Tx
5 -----	5	Gnd

El entrecruzamiento entre los pines Tx y Rx se lleva a cabo en el circuito de la tarjeta DSP, razón por la cual, en el cable de comunicación serial, las líneas se conectaron de la manera ilustrada.

Se comprobó el correcto funcionamiento del circuito integrado MAX 3238CPW en la tarjeta DSP con un osciloscopio conectado a las salidas ROUT1(pin 21) y DIN1 (pin 24) del integrado en el momento en que recibe un dato desde el PC (ver archivo ti\_ezdsf2808usb.opj) incluido en el CD de code composer studio.

La conexión entre las tarjetas 2 y 3 se muestra en la figura 8.



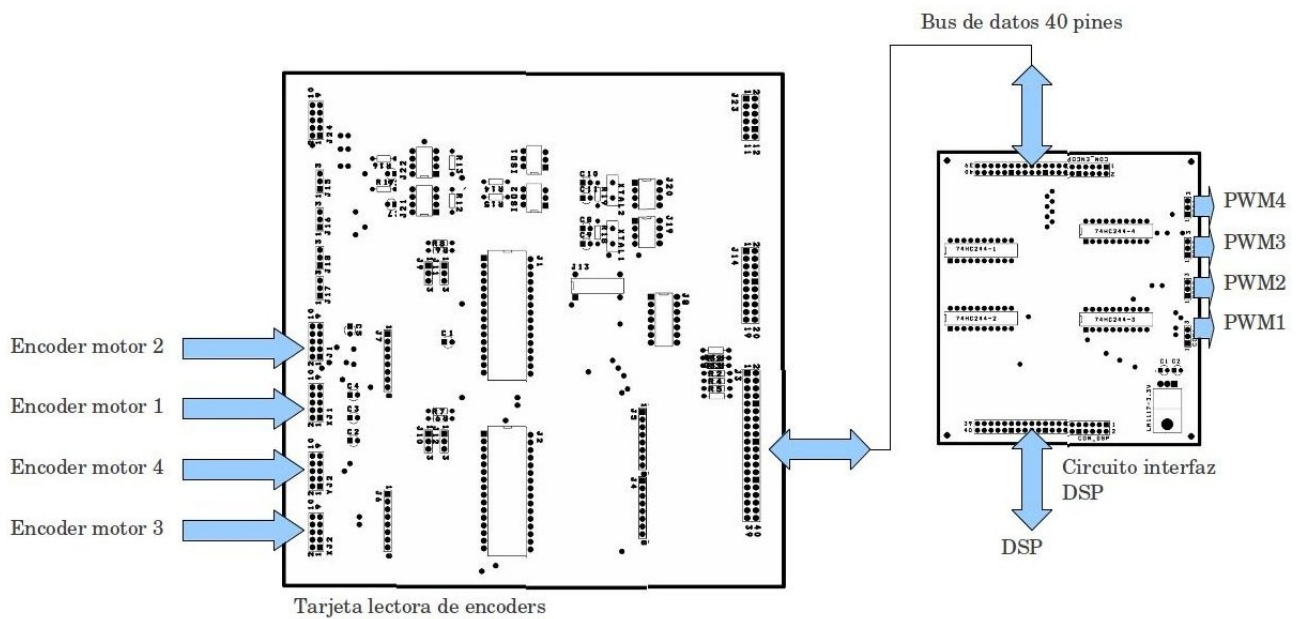


Figura 8.

El bus de 40 pines originalmente tenía una longitud de 1 metro, por lo que fué recortado, ajustándolo a la distancia entre las tarjetas.

La conexión de las tarjetas 2, 4, 5 y 6 se muestran en la figura 9.

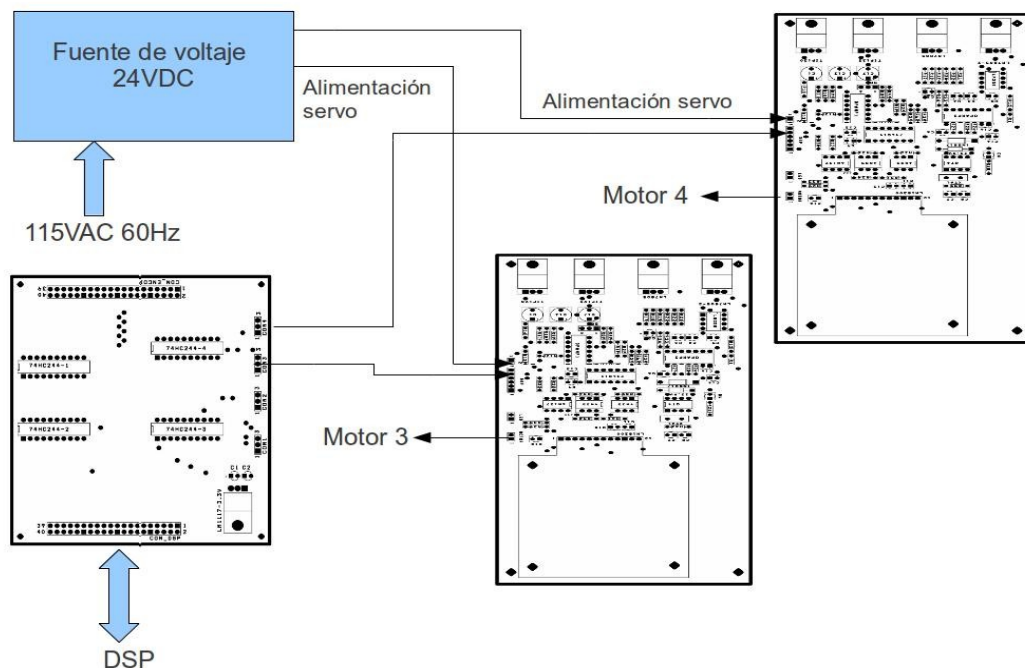


Figura 9.

#### INSTALACIÓN Y PRUEBA DEL CODE COMPOSER STUDIO

Una vez finalizada la conexión del hardware se procede a la revisión del software almacenado en la memoria flash de la tarjeta de desarrollo del DSP, para lo cual se instala el programa code composer studio V3.1. Éste programa viene junto con la tarjeta de desarrollo del DSP TMS320F2808. Éste CD ROM también trae los drivers para la comunicación a través del puerto USB entre el PC y el DSP.

Después de instalar el code composer studio y los drivers, se inicia el programa Setup CCStudio V3.1.exe, en el cual seleccionamos la plataforma de desarrollo F2808 eZdsp y escojemos “save & Quit”, (figura 10.)

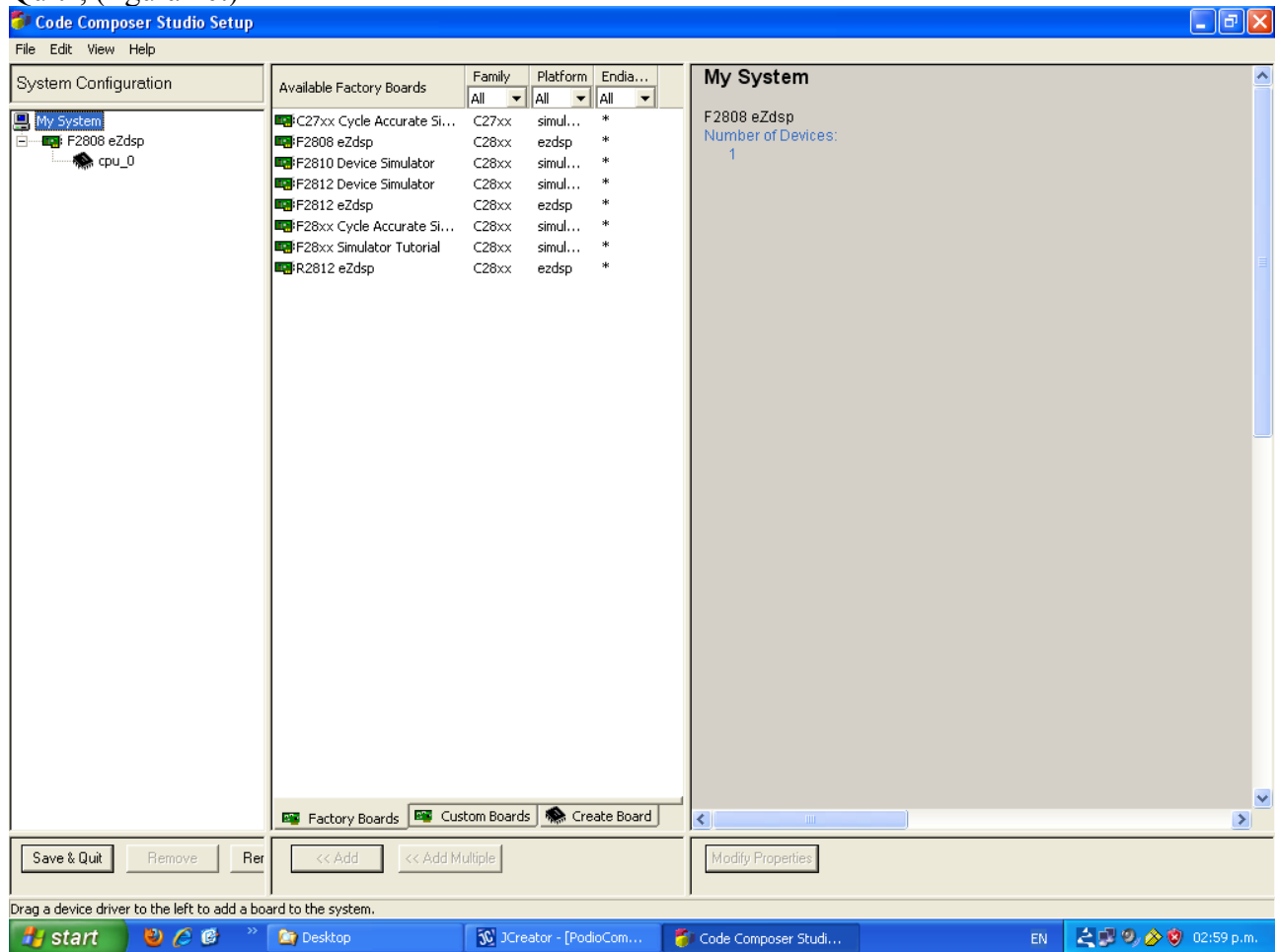


Figura 10.

Ésto crea un archivo ejecutable llamado F2808 eZdsp CCStudio V3.1.exe, que corresponde al entorno para el desarrollo de aplicaciones en el DSP.

Después de iniciar este programa, y de conectar el DSP al computador por el puerto USB, para nuestro caso, se requiere conocer el estado del programa en el DSP, por lo que se elige Debug → connect, (figura 11).

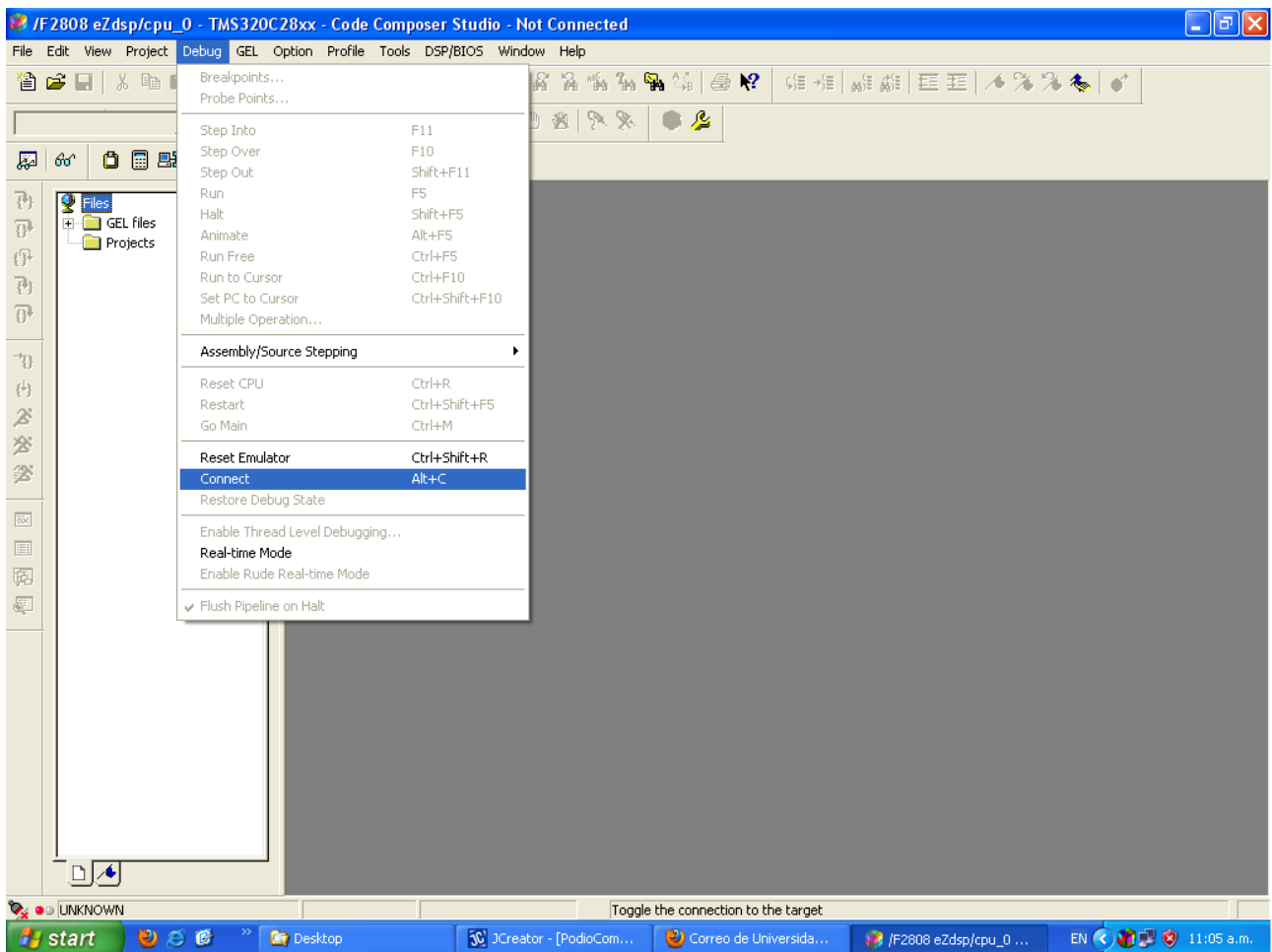


Figura 11.

esto abre una ventana en donde se muestra el programa residente en la memoria del DSP pero en lenguaje ensamblador, (figura 12) lo que nos indica que el DSP sí tiene un programa almacenado en la memoria Flash.

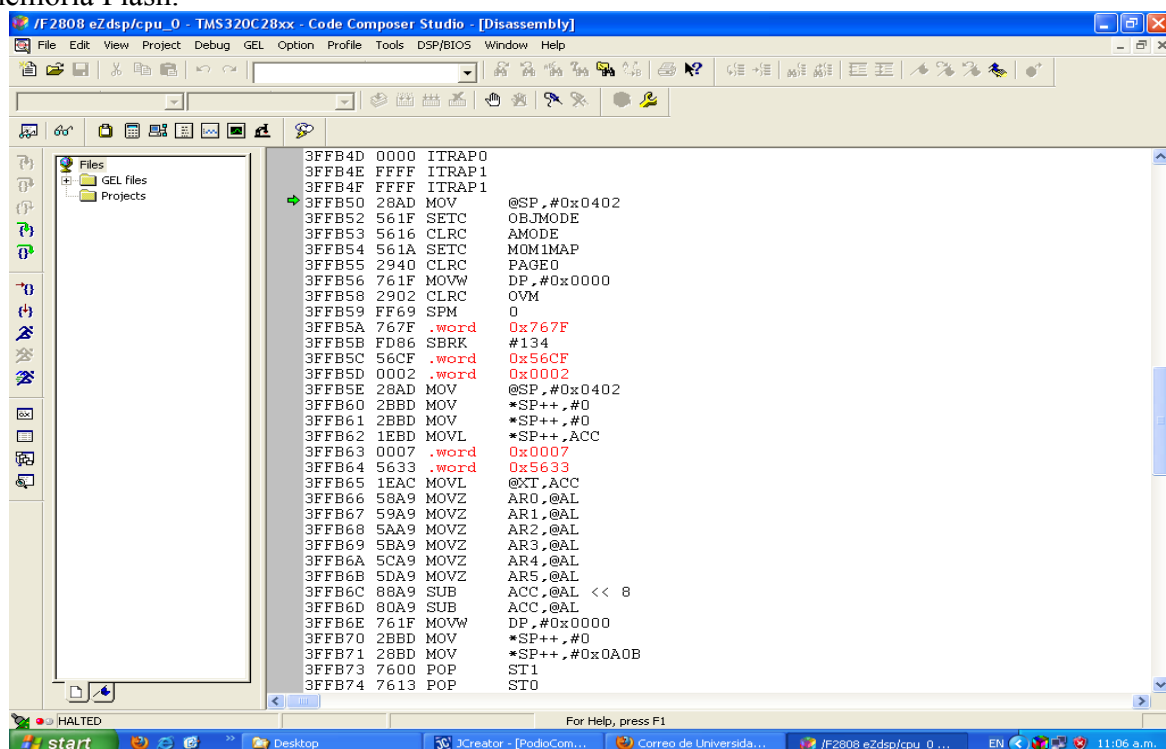


Figura 12.



NOTA: Esta prueba se realizó después de haber leído el manual 2808\_ezdspusb\_techref\_c.pdf incluido en el CD\_ROM del code composer, en el cual se especifica que el boot por memoria flash se logra mediante la configuración del dipswitch nombrado SW1:

Switch 1-3 OPEN

Una vez cargado el programa en assembler, no fué posible realizar la conversión a lenguaje C, ya que según Spectrum Digital, no existe un programa que realice tal traslado, y llevar a cabo un análisis del código en assembler tomaría mucho tiempo.

Con esta prueba se pudo verificar que el DSP contiene un programa, y según registros, correspondería al programa de control del podio hecho en el 2007.

### EJECUCIÓN DE LA APLICACIÓN DE CONTROL DEL PODIO EN JAVA

Se encontraron dos copias del programa en JAVA realizado para el control del podio con joystick, una copia se encuentra en el disco duro del computador IBM Intellistation del proyecto y otra copia en uno de los discos compactos entregados por el equipo que desarrolló el proyecto en el 2007, a su vez se encontraron 2 versiones de esa aplicación llamadas PodioComm y PodioComm2 (figura 13) con algunas diferencias notables, entre las cuales se nombra la más significativas:

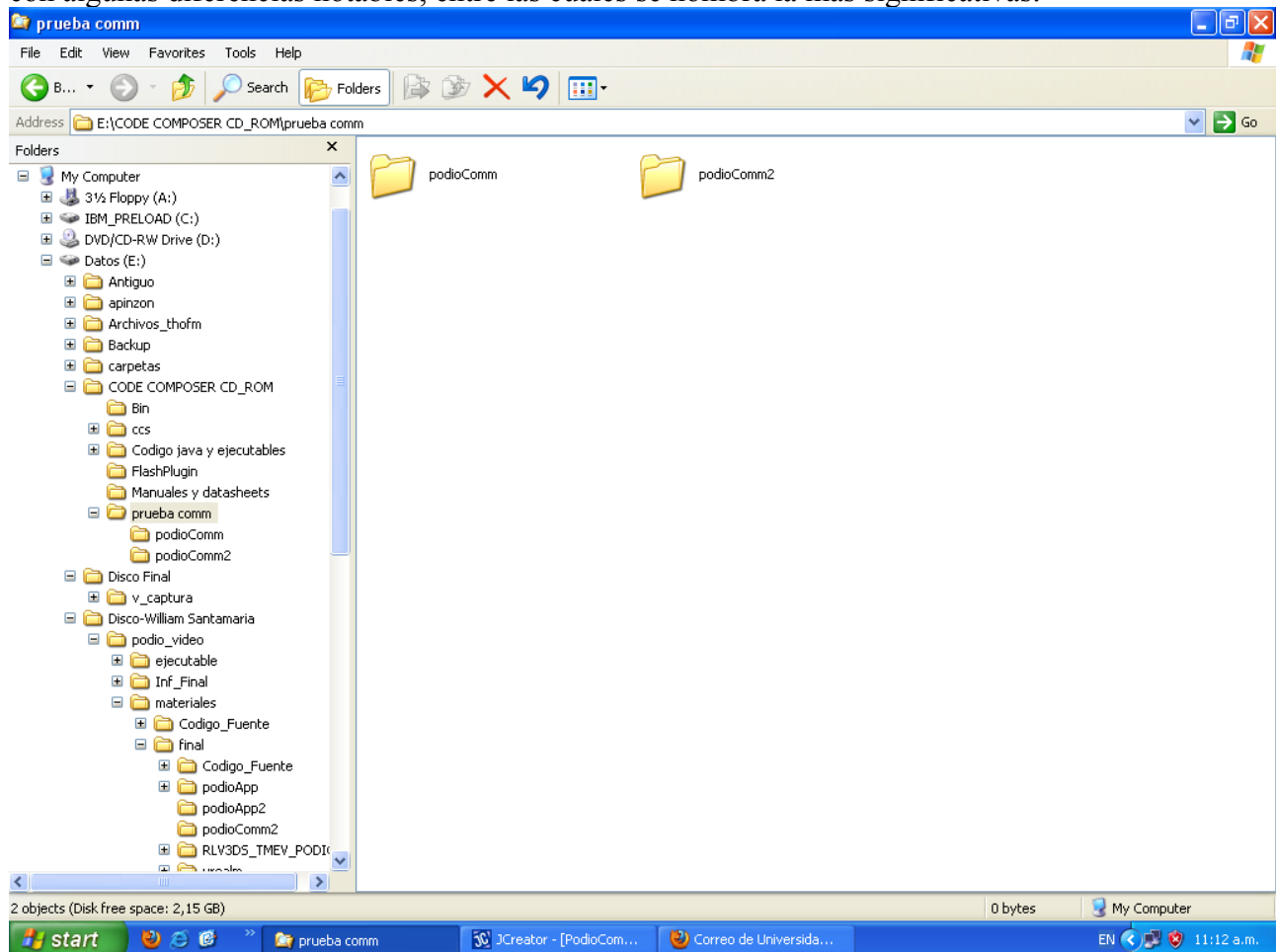


Figura 13.

PodioComm

Configura la comunicación serial con el DSP a 38400 bps

Mientras que la configuración de la aplicación PodioComm2 es la siguiente:

Configura la comunicación serial con el DSP a 9600 bps

La configuración del dato enviado al DSP es:

Formato: {start, j1, 0.0, stop,}

donde:

Start → inicia el movimiento

j1 → selecciona el motor (o articulación, en este caso motor 1),

0.0 → la velocidad y dirección

con valores entre 0.1 y 1 para velocidad en dirección FWD

con valores entre -0.1 y -1 para velocidad en dirección RWD

0.0 no hay movimiento

Stop → Detiene el movimiento

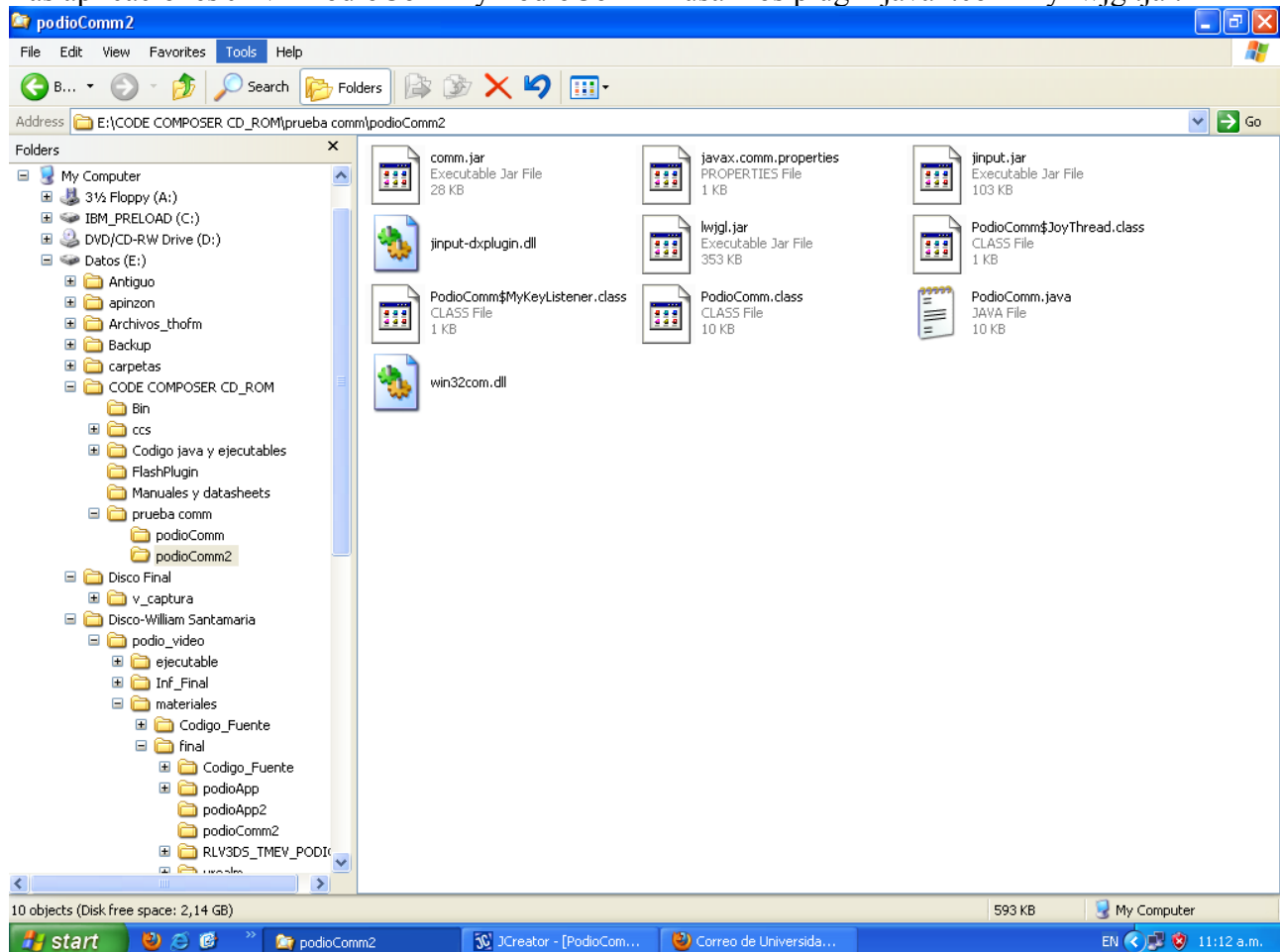
La trama de recepción de datos es la siguiente:

Formato: {start, A1, A2, A3, A4, stop}

donde Ax → corresponde al ángulo de cada respectiva articulación.

Se desconoce la configuración del puerto serial en el programa del DSP, por lo que se tuvo que realizar la prueba de comunicación con ambas aplicaciones (PodioComm y PodioComm2), pero antes de poder correr éstos programas se tuvo que recurrir a la siguiente configuración:

Las aplicaciones JAVA PodioComm y PodioComm2 usan los plugin javax.comm y lwjgl.jar.



Javax.comm → éste plugin se encarga de la comunicación serial y consta de 3 archivos:

**win32com.dll**

**comm.jar**

**javax.comm.properties**

éstos archivos se pueden descargar de internet y se deben copiar tanto para JDE como para JDK de la siguiente forma:

copiar el archivo win32comm.dll y pegarlo en:

C:\Program\_Files\Java\jre1.5.0\_22\bin\  
C:\Program\_Files\Java\jdk1.5.0\_22\bin\  
C:\WINDOWS\system32\

copiar el archivo comm.jar y pegarlo en:

C:\Program\_Files\Java\jre1.5.0\_22\lib\ext\  
C:\Program\_Files\Java\jdk1.5.0\_22\lib\  
C:\Program\_Files\Java\jdk1.5.0\_22\lib\ext

copiar el archivo javax.comm.properties y pegarlo en:

C:\Program\_Files\Java\jre1.5.0\_22\lib\  
C:\Program\_Files\Java\jdk1.5.0\_22\lib\  
C:\Program\_Files\Java\jdk1.5.0\_22\jre\lib

El path para el programa java jre y jdk puede variar si se utiliza en otro computador, ésta información se obtuvo de: <http://lefunes.wordpress.com/2009/02/27/instalacion-del-java-comm-api-en-windows/>

El plugin lwjgl se encarga de la lectura de los movimientos del joystick y consta de tres archivos:

**lwjgl.jar**  
**jinput.jar**  
**jinput-dxplugin.dll**

No se encontró información acerca de la ubicación exacta de cada archivo, por lo que basado en el anterior caso, se procedió de la siguiente manera:

Copiar los archivos **lwjgl.jar**, **jinput.jar** y pegarlos en:

C:\Program\_Files\Java\jre1.5.0\_22\bin\  
C:\Program\_Files\Java\jdk1.5.0\_22\bin\  
C:\Program\_Files\Java\jre1.5.0\_22\lib\ext\  
C:\Program\_Files\Java\jre1.5.0\_22\lib\  
C:\Program\_Files\Java\jdk1.5.0\_22\lib\  
C:\Program\_Files\Java\jdk1.5.0\_22\lib\ext  
C:\Program\_Files\Java\jdk1.5.0\_22\jre\lib

Copiar el archivo **jinput-dxplugin.dll** y pegarlo en:

C:\Program\_Files\Java\jre1.5.0\_22\bin\  
C:\Program\_Files\Java\jdk1.5.0\_22\bin\  
C:\WINDOWS\system32\

Una vez realizado, se abre el archivo PodioComm.java con Jcreator Pro version 3.10.008, (Figura 15).

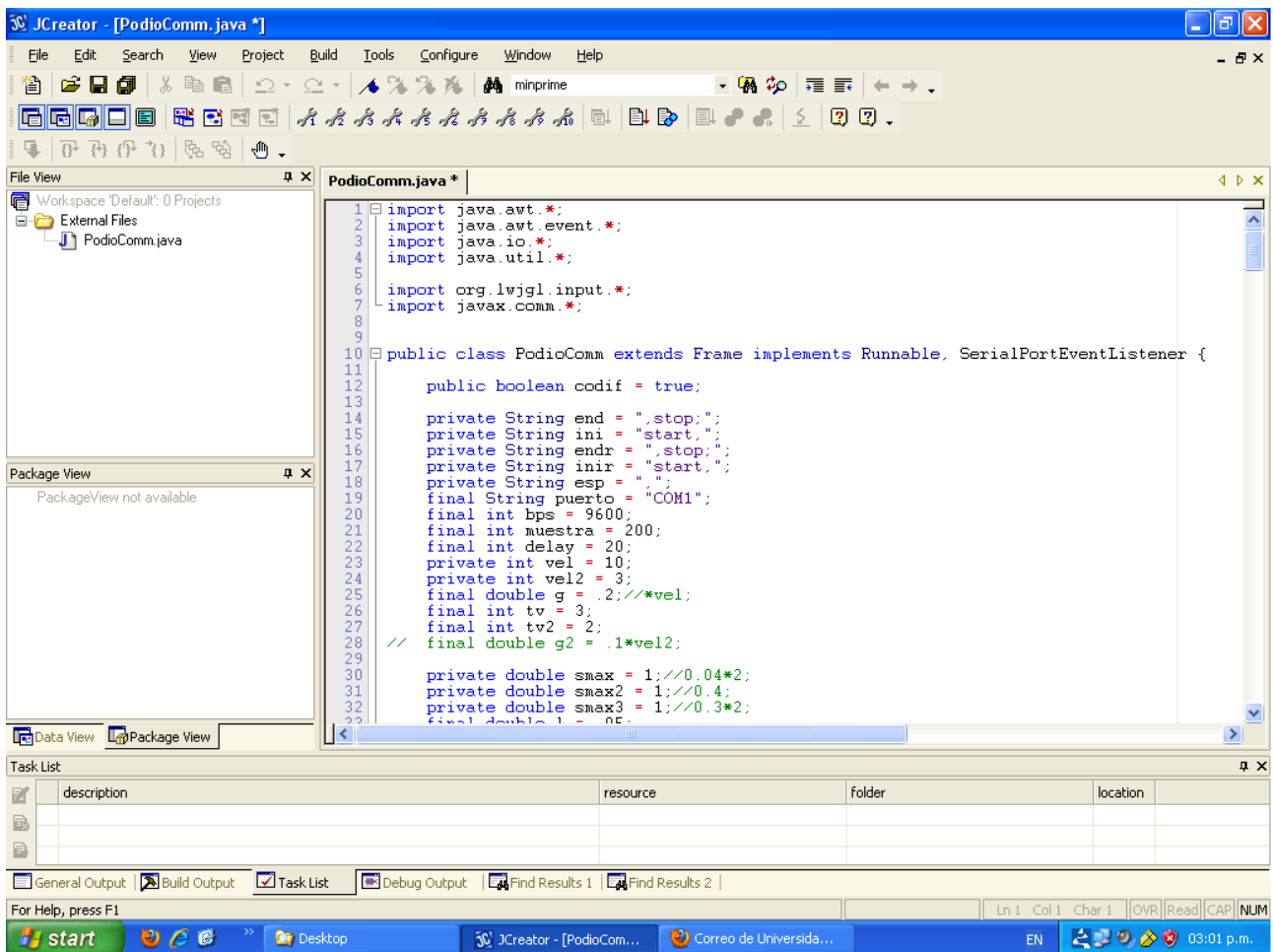


Figura 15.

NOTA: para que la aplicación funcione, es obligatorio tener conectado el joystick en el puerto USB. Después en Build → compile se realiza la compilación del programa (Figura 16).

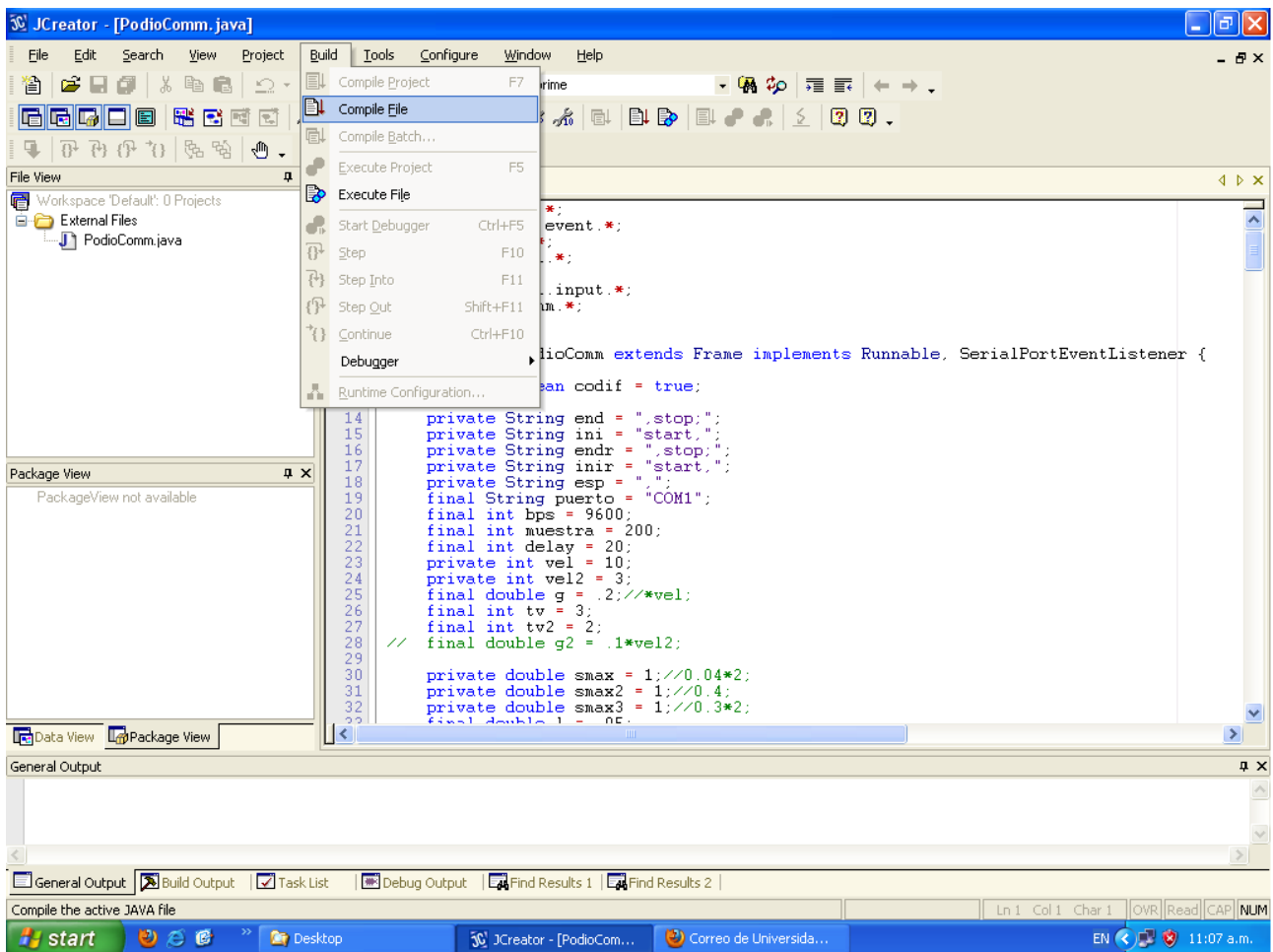


Figura 16.

Si la compilación es exitosa aparece el debe aparecer un mensaje “Process completed” en el recuadro “General Output”. (figura 17).

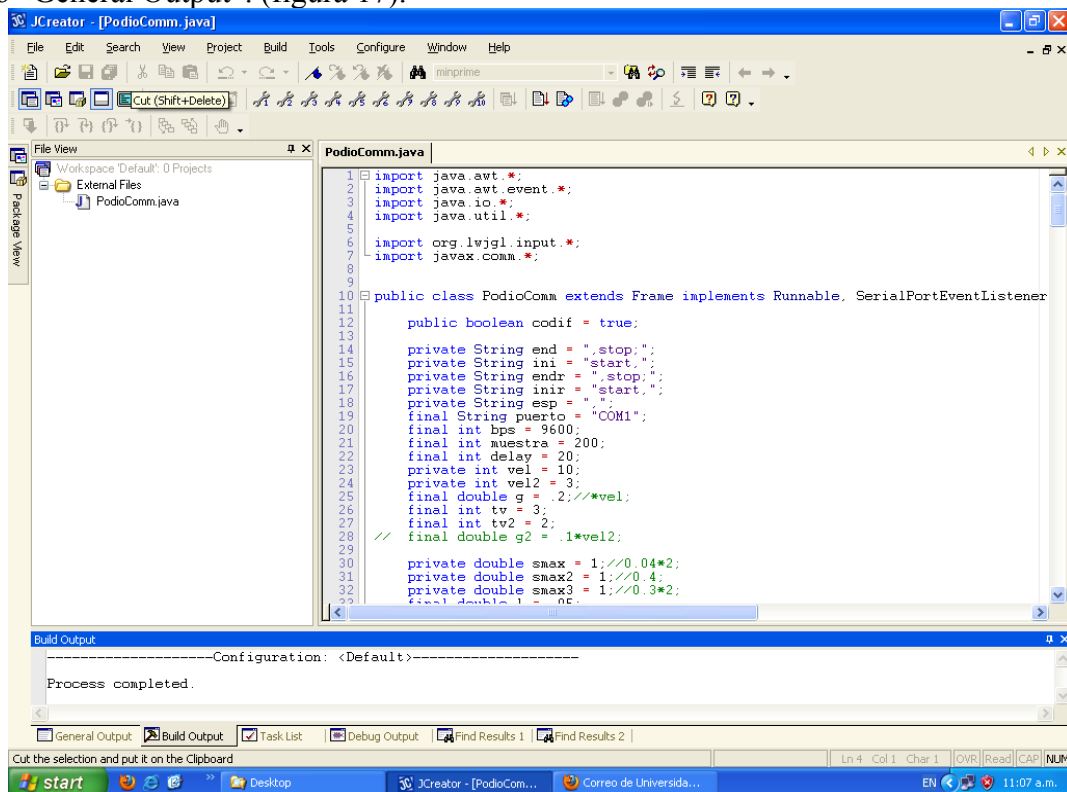


Figura 17.

The screenshot shows the JCreator IDE with the 'Execute File' menu option selected. The main editor displays the following Java code:

```

14 private String end = ".stop.";
15 private String ini = ".start.";
16 private String endr = ".stop.";
17 private String inir = ".start.";
18 private String esp = ".";
19 final String puerto = "COM1";
20 final int bps = 9600;
21 final int muestra = 200;
22 final int delay = 20;
23 private int vel = 10;
24 private int vel2 = 3;
25 final double g = .2;//*vel;
26 final int tv = 3;
27 final int tv2 = 2;
28 // final double g2 = .1*vel2;
29
30 private double smax = 1;//0.04*2;
31 private double smax2 = 1;//0.4;
32 private double smax3 = 1;//0.3*2;
33 final double l = .05;

```

The Build Output window shows the following message:

```

-----Configuration: <Default>-----
Process completed.

```

The status bar at the bottom indicates that the 'General Output' tab is selected, and the 'Build Output' tab is also visible. The system tray shows the Windows Start button, the JCreator icon, and the system clock displaying 11:08 a.m.

Luego aparece la aplicación, en donde se aprecia los comandos enviados a través del puerto serial: (figura 19)





#### Verificaciones realizadas:

La transmisión de datos desde el PC hacia el DSP se verificó con un osciloscopio conectado en la tarjeta DSP según planos encontrados en el archivo 2808\_ezdspusb\_schem\_a.pdf incluido en el CD\_ROM del code composer.

Se verificó en 5VDC el voltaje de alimentación de la tarjeta lectora de encoder, así como de los encoder y la correcta polarización de la tarjeta interfaz DSP.

Se verifica el voltaje de 24VDC en los circuitos servoamplificadores.

Se verifica la correcta conexión de los pines de los encoder y la totalidad de las señales.

#### Resultados de las pruebas:

Con la comunicación serial establecida, el podio no realiza ningún movimiento.

Al inspeccionar los pines de salida del DSP en donde debería verse una trama de pulsos (PWM hacia la tarjeta servoamplificadora), no se observa ninguna señal, lo que indica que el DSP no está ejecutando las instrucciones recibidas desde el PC.

Al inspeccionar los datos recibidos desde el circuito decodificador de cuadratura, no se encuentra ningún dato, dado que los pines de control de los circuitos integrados HCTL 2032 llamados OE, SEL1, SEL2, RST, XY, son controlados por el DSP y éste no está ejecutando ninguna acción.