

Bogotá 19 de Febrero del 2012

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

ENTREGABLE #3

César Augusto Pantoja

Estudiante auxiliar ingeniería electrónica

Programación del DSP

Introducción

El podio cuenta con una tarjeta DSP ya programada, sin embargo no se conocen las razones por las cuales no responde a los comandos enviados desde la aplicación de control en Java 3D, por lo cual se opta por iniciar un nuevo programa en otra tarjeta DSP de la misma referencia que se encuentra disponible.

Para programar la tarjeta DSP TMS320F2808 de digital Spectrum, se utiliza el programa Code Composer Studio V3.1 el cual viene en un CD_ROM incluido en la caja del DSP. El software permite la programación del DSP en C++, así como el Debug y la descarga hacia la memoria sram o a la memoria FLASH. También existe la opción de crear el programa desde la herramienta de matlab, “Simulink” la cual utiliza a su vez otra herramienta llamada “Embebbbed coder” en la cual se encuentra las librerías C2000 quien contiene los módulos del DSP de texas instruments.

A continuación se muestra los pasos para la creación y configuración de un programa que se encarga de efectuar movimientos básicos en el podio.

1. Configuración de los equipos:

Code Composer Studio V3.1 no es compatible con matlab, ni es compatible con Windows 7, por lo cual se procede a descargar una versión gratuita de Code Composer Studio V3.3 MCU, la cual es compatible con Windows Vista, pero se ensayó en windows 7 Starter y funciona perfectamente; se diferencia de la versión completa en que no cuenta con los drivers para los emuladores de la mayoría de las tarjetas, por lo que fué necesario descargar los drivers para CCS V3.3 de:

<http://support.spectrumdigital.com/>. Después click en 'Installation under CCS 3.3'.

Después de descargar el driver, se hace doble click y se siguen las instrucciones de instalación.

Matlab y Code Composer V3.3 deben estar instalados en el mismo equipo.

Entonces:

CCS V3.3MCU y los drivers para la board F2808 instalados.

Matlab r2010b instalado con todos los tools.

Windows 7 Starter

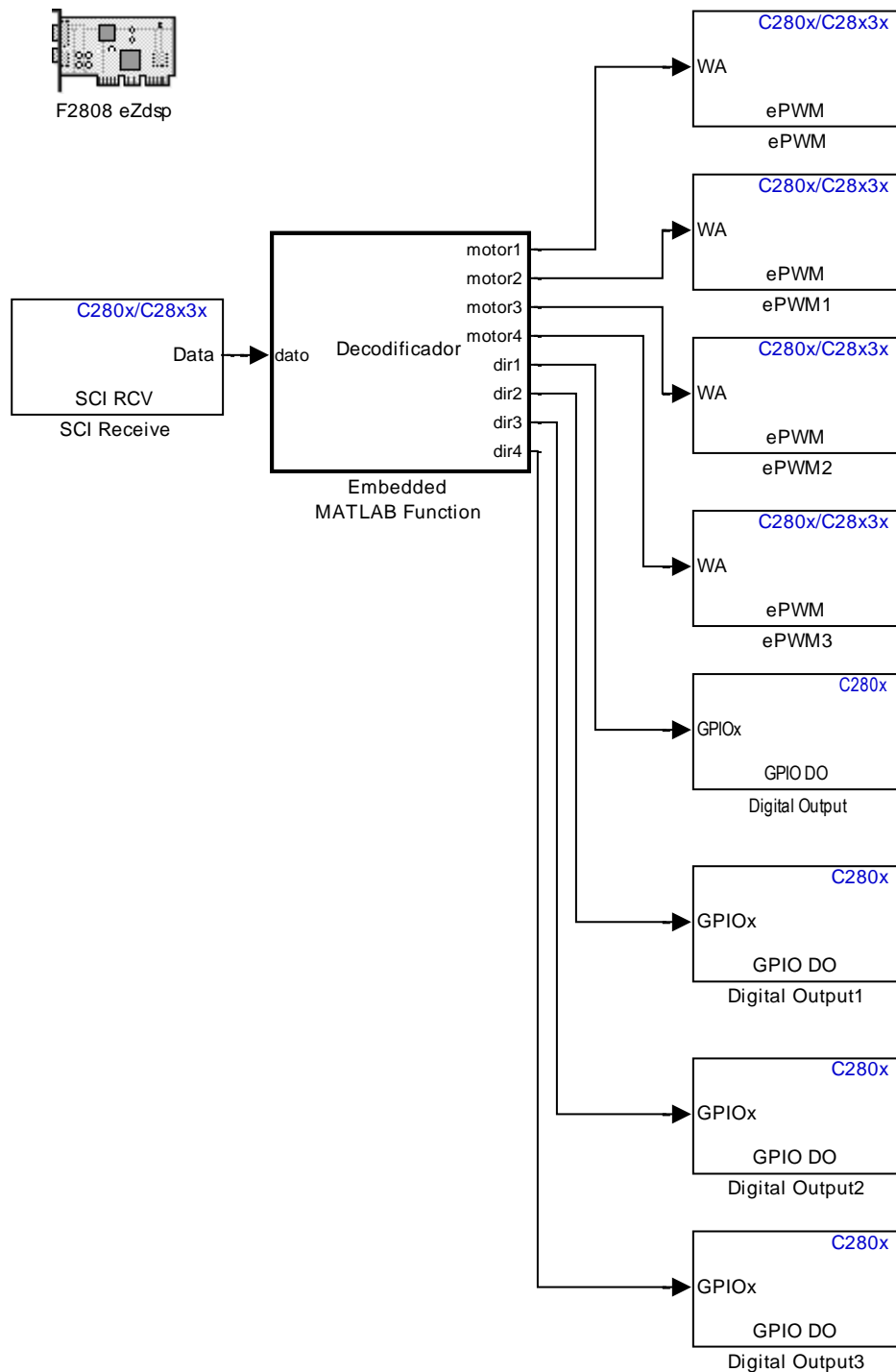
El procedimiento para hacer el programa es la siguiente:

- 1.1 Hacer click con el botón derecho sobre el ícono de Matlab y elegir la opción “Ejecutar como Administrador”.
- 1.2 Escribir en la consola de matlab “simulink”
- 1.3 Una vez abierto simulink ir a Archivo luego nuevo.
- 1.4 En simulink hacer click en “Target Supported” , luego en C2000 y luego en C280X.
- 1.5 Luego se arrastra el ícono F2808 hacia la hoja en blanco

Al hacer esto se abre un cuadro de diálogo que informa que se cargará una configuración para la board F2808, así que se da click en aceptar. Con lo cual ya queda el espacio para introducir los

bloques para el programa.

Programa para control de los motores a través del puerto serial.



El bloque SCI Receive se encarga de leer el puerto serial, y entrega los datos en forma de vector hacia el bloque decodificador.

El bloque llamado Decodificador, se encarga de leer el dato obtenido del puerto serial y ejecutar el accionamiento del eje correspondiente, el dato obtenido desde el puerto serial corresponde a un vector de 3 elementos, distribuidos así:

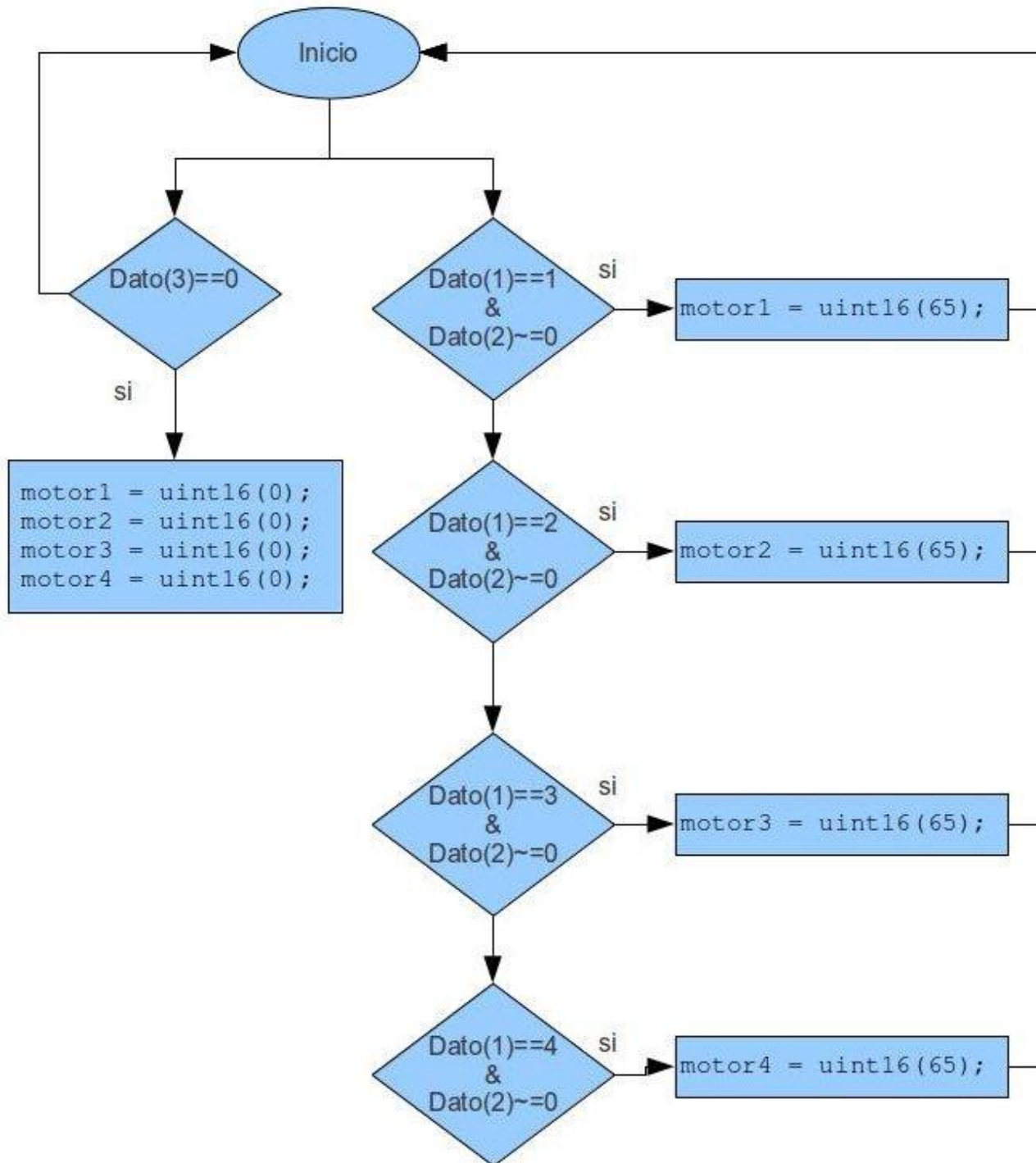
Dato[eje, direccion, velocidad] en donde:

eje → 1, 2, 3, 4

Velocidad → 0 hasta 100

Dirección → 0, 1

así que a partir de esta información se construye el siguiente diagrama de flujo:



A partir del diagrama de flujo, se crea el bloque Decodificador con el siguiente código, visible al hacer doble click en el mismo bloque:

```
function [motor1,motor2,motor3,motor4,dir1,dir2,dir3,dir4] = Decodificador(dato)
%#eml

% Control movimiento articulacion 1
if (dato(1) == '1' & dato(2) ~= '0')
    motor1 = uint16(65);
else
```

```

        motor1 = uint16(0);
end

if (dato(3) == '0')
    dir1 = uint16(0);
else
    dir1 = uint16(1);
end

% Control movimiento articulacion 2
if (dato(1) == '2' & dato(2) ~= '0')
    motor2 = uint16(65);
else
    motor2 = uint16(0);
end

if (dato(3) == '0')
    dir2 = uint16(0);
else
    dir2 = uint16(1);
end

% Control movimiento articulacion 3
if (dato(1) == '3' & dato(2) ~= '0')
    motor3 = uint16(50);
else
    motor3 = uint16(0);
end

if (dato(3) == '0')
    dir3 = uint16(0);
else
    dir3 = uint16(1);
end

% Control movimiento articulacion 4
if (dato(1) == '4' & dato(2) ~= '0')
    motor4 = uint16(15);
else
    motor4 = uint16(0);
end

if (dato(3) == '0')
    dir4 = uint16(0);
else
    dir4 = uint16(1);
end

```

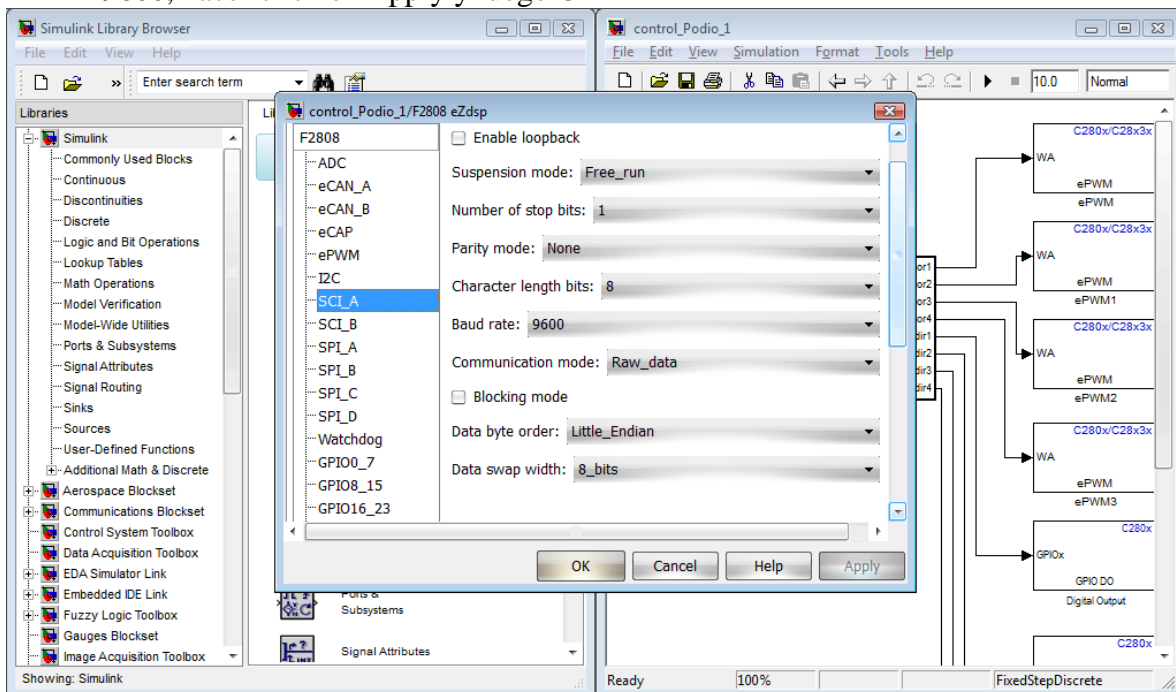
El bloque tiene como salidas una señal de porcentaje de PWM hacia el bloque PWM y la dirección la cual corresponde a un “1” ó un “0” hacia un GPIO quien finalmente la envía hacia la tarjeta servoamplificadora encargada de gestionar el movimiento.

El bloque PWM se encarga de crear las tramas de ancho de pulso variable según el porcentaje recibido desde el bloque decodificador.

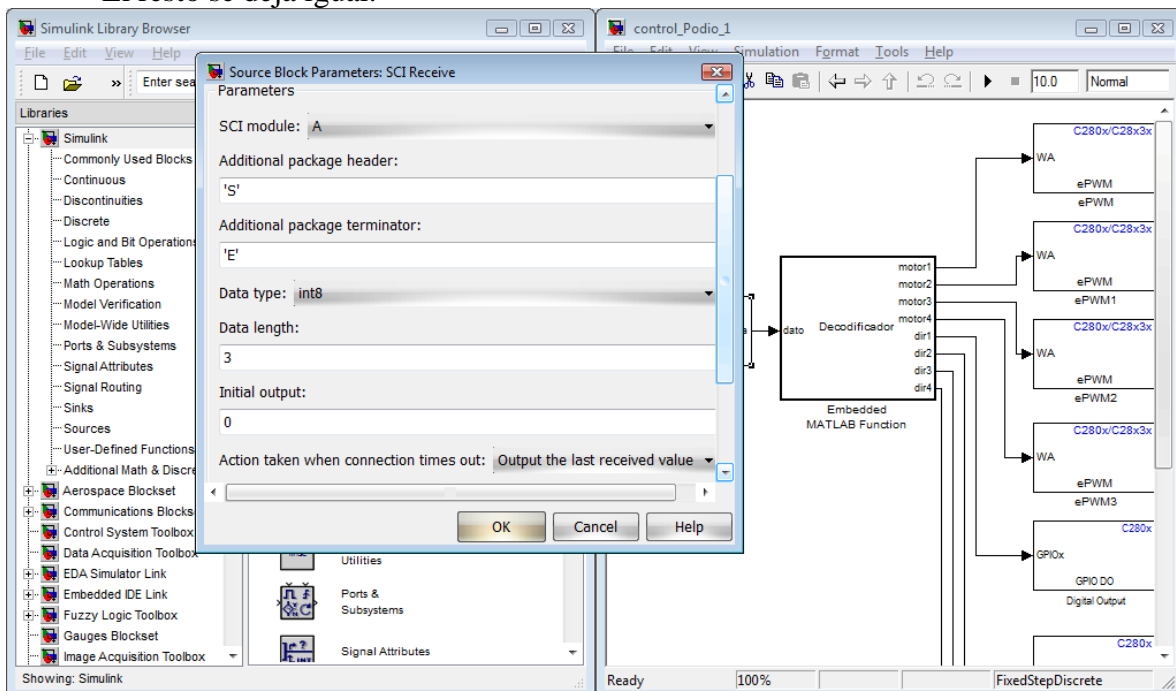
El bloque GPIO es un bloque encargado de manejar un pin físico del DSP, con el parámetro obtenido del bloque decodificador, éstos pines se encuentran conectados físicamente al pin dir de la tarjeta servoamplificadora.

CONFIGURACIONES DEL PROGRAMA:

1. Hacer doble click en el bloque F2808 y luego en SCI_A y cambiar Baud rate de 115800 a 9600, hacer click en Apply y luego OK



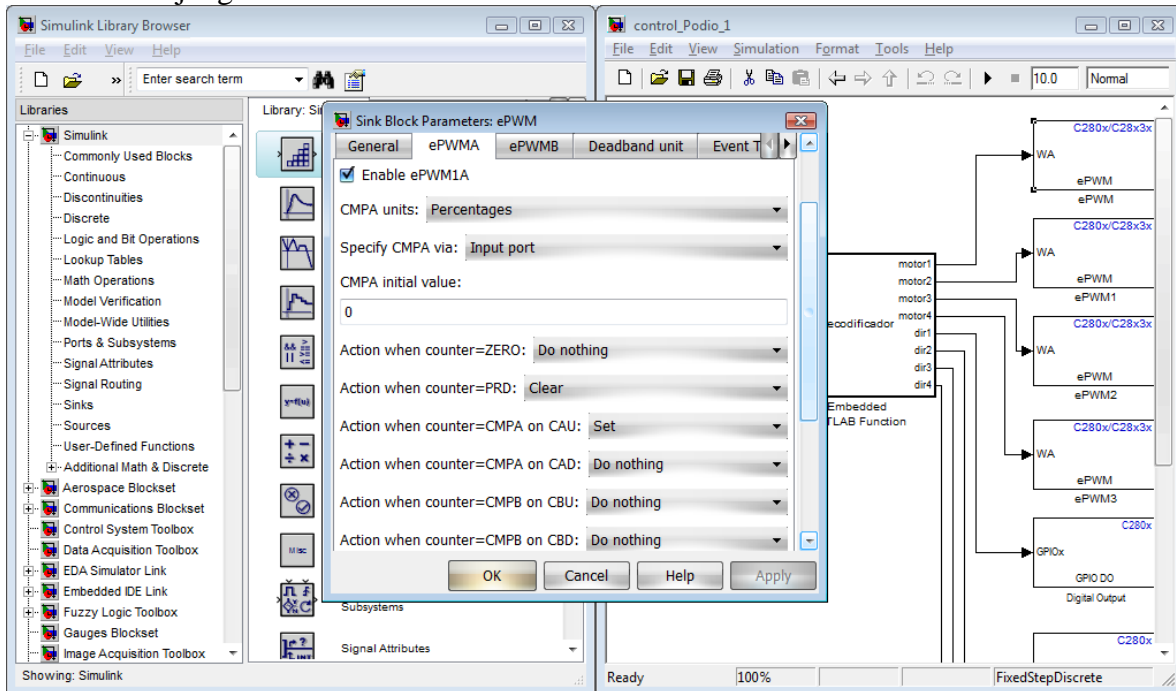
2. Hacer doble click en el bloque SCI_RCV y configurar parámetros así:
Data Type : int8
Data Length: 3
El resto se deja igual.



Hacer doble click en cada uno de los bloques PWM y configurar lo siguiente:
casilla Enable ePWMA habilitada
CMPA units: Percentages
Specify CMPA vía: Input Port

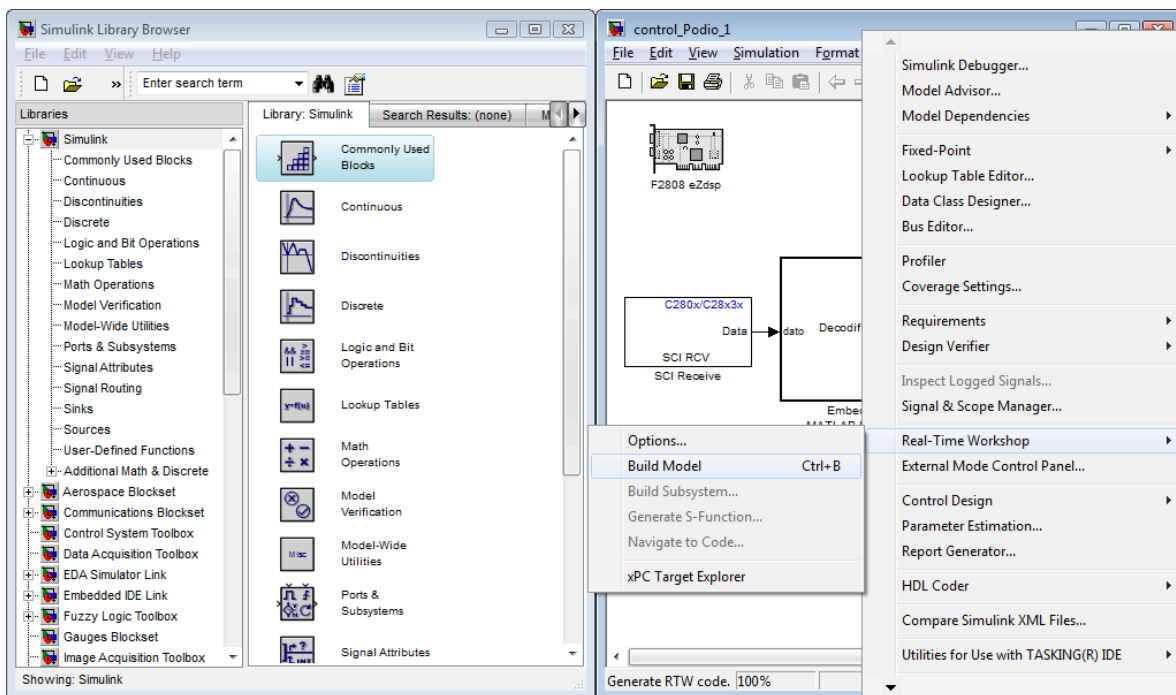
CMPA initial value: 0

El resto se deja igual

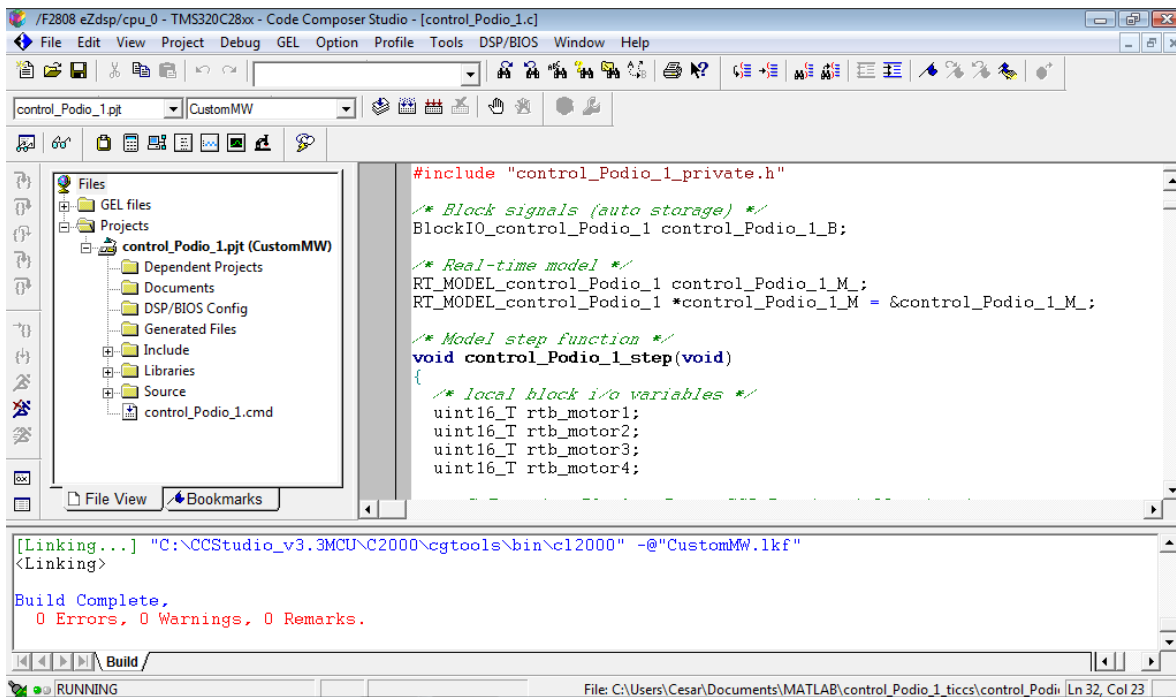


COMPILACION Y GENERACIÓN DEL CODIGO PARA EL CODE COMPOSER STUDIO

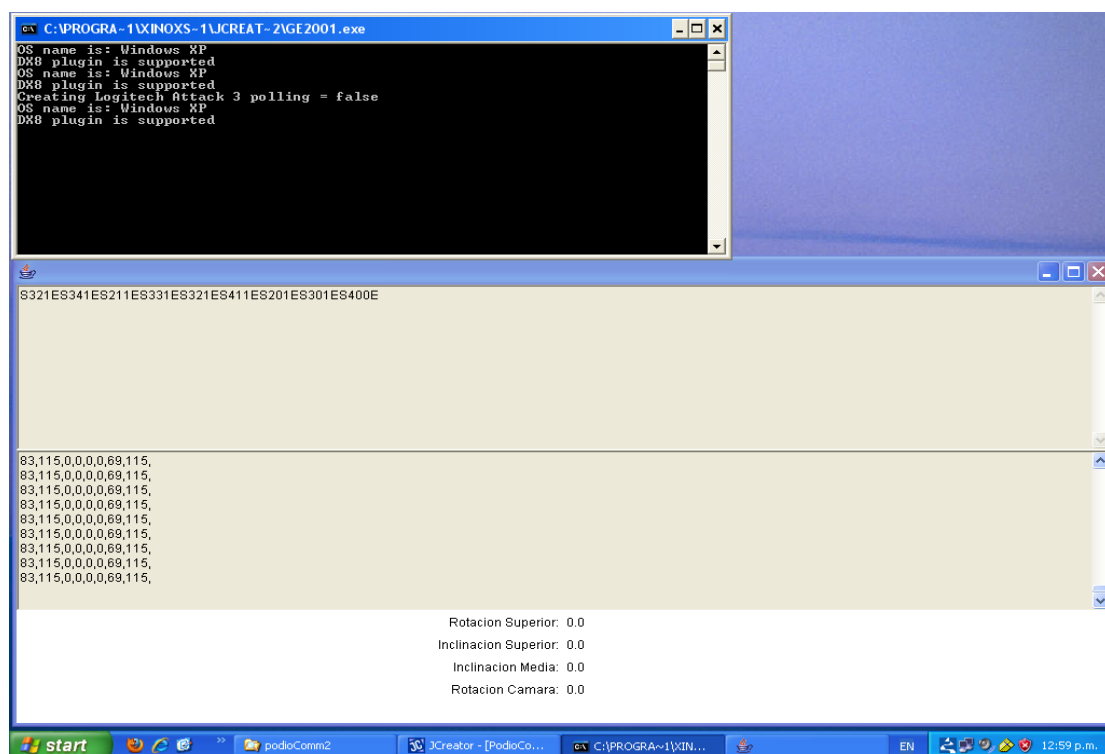
Una vez el programa se encuentra terminado hacemos clic en la pestaña tools, luego en Real-Time Workshop y finalmente en Build model:



Si todo resulta sin errores, se abre automáticamente la ventana del Code Composer Studio en donde se compilan los codigos generados por simulink y al final se carga el programa en la memoria sram del DSP.



La aplicación ya esta cargada en el DSP, ahora procedemos a abrir la aplicación java de prueba, teniendo el joystick conectado al equipo:



CONCLUSIONES

1. El podio responde bien al programa, aunque falta añadir el generador de trayectorias.
2. La creación de la aplicación fué relativamente sencilla, pero la configuración dió bastantes problemas, fué necesario hacer muchos ensayos antes de dar con la más óptima.
3. Versiones mas modernas del code composer studio se encuentran disponibles en versiones gratuitas, sin embargo se utilizó la versión 3.3 porque es la que mejor desempeño mostró al usarse en conjunto con simulink de matlab.

4. Toda la configuración debe hacerse tal cual para que el programa sea compatible con la aplicación Java 3D de control.
5. Los motores están presentando síntomas de falta de torque, está pendiende determinar las causas.

