

Bogotá 18 de Marzo del 2012

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

ENTREGABLE #4

César Augusto Pantoja

Estudiante auxiliar ingeniería electrónica

**INSTALACIÓN DE DOS TARJETAS SERVOAMPLIFICADORAS PWM Z12A8DDC  
PARA LOS MOTORES DE LOS EJES 3 Y 4**

Se recibieron dos tarjetas PWM SERVO AMPLIFIERS

Models: Z12A8DDC

Micro Series

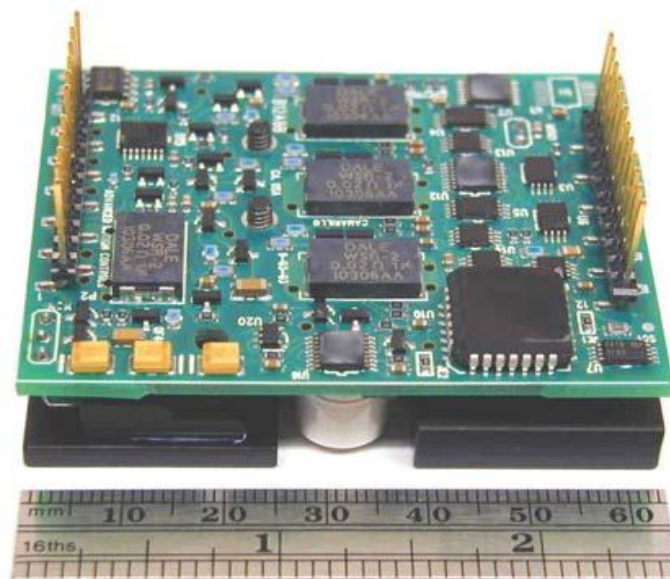


Figura 1.

**INSTALACIÓN**

Para la instalación se diseñó un circuito impreso (PCB) para la adaptación de los conectores utilizados en los otros módulos PWM y facilitar el intercambio, según se requiera. El circuito diseñado

aparece en la figura 2.

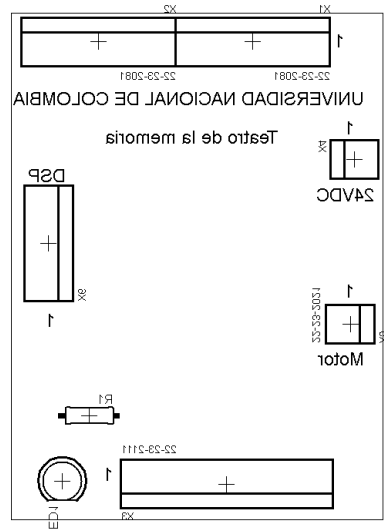
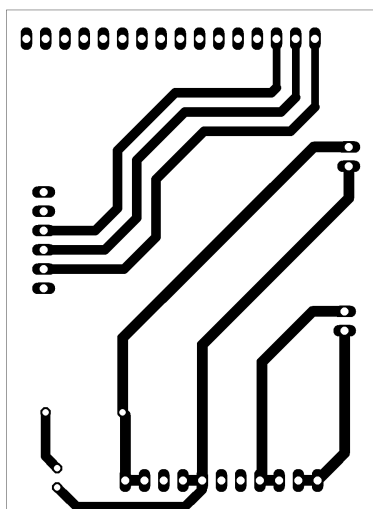


Figura 2.

En la figura 2 se muestran el diseño de las pistas (lado izquierdo) y el diseño del lado de los componentes,

La pcb terminada junto con la PCB del circuito PWM servoamplifier Z12A8DDC (Figura 3)

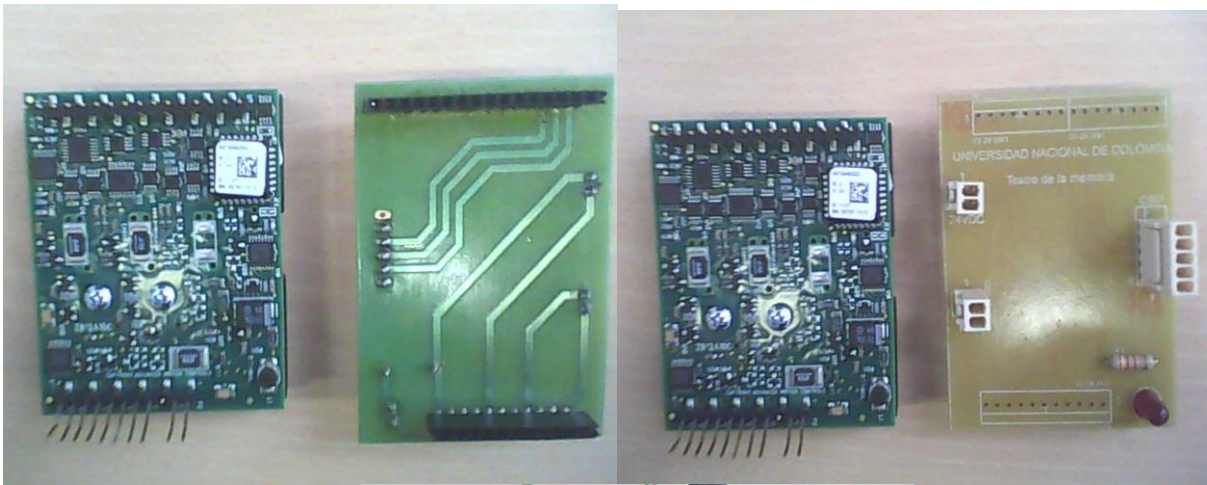


Figura 3.

Las conexiones (figura 4) corresponden con el diagrama de conexiones extraídas del datasheet del PWM servoamplifier Z12A8DDC

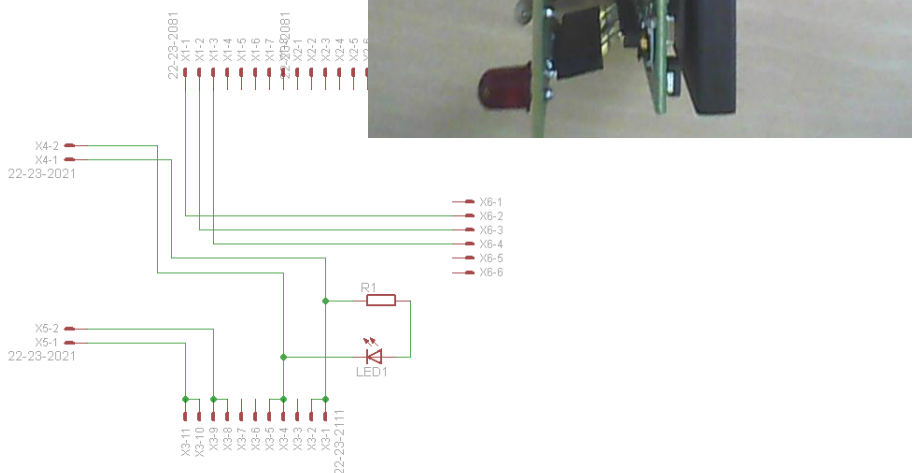


Figura 4.

Básicamente se conectaron las señales necesarias, pwm, direction, Gnd, +24VDC, GND, motor. Siguiendo el diagrama esquemático extraído 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

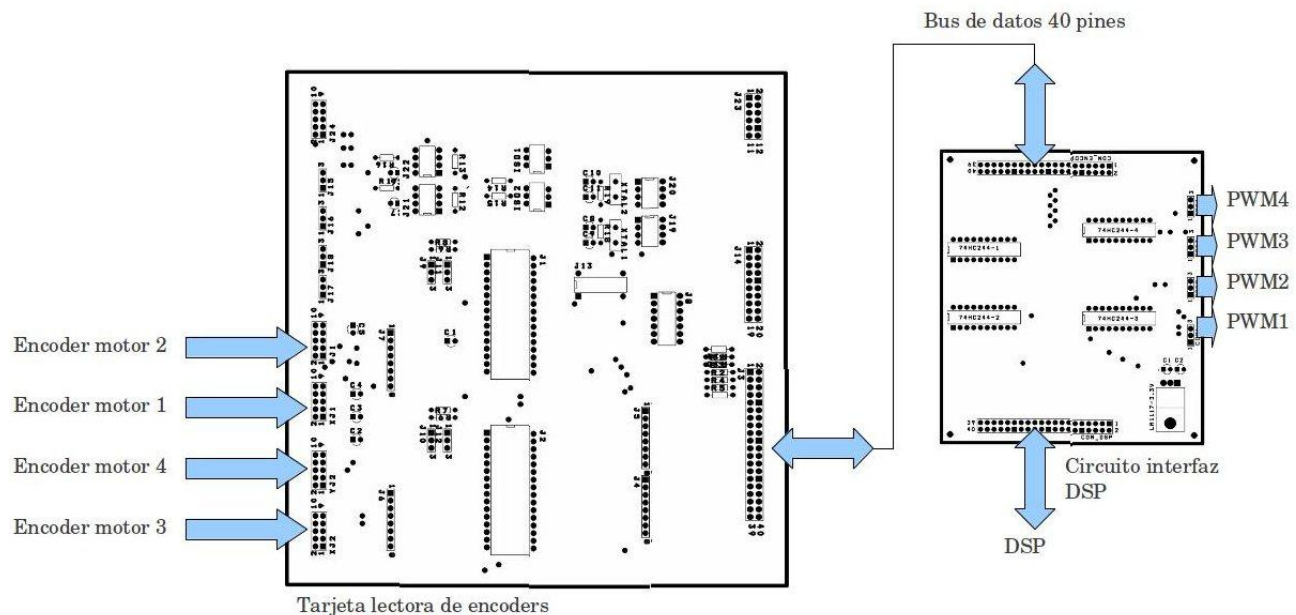


Figura 5.

El circuito llamado “Circuito interfaz DSP” (figura 5) es el encargado de enviar las señales PWM, dir, Gnd hacia la tarjeta servoamplificadora Z12A8DDC.

#### PRUEBA REALIZADA

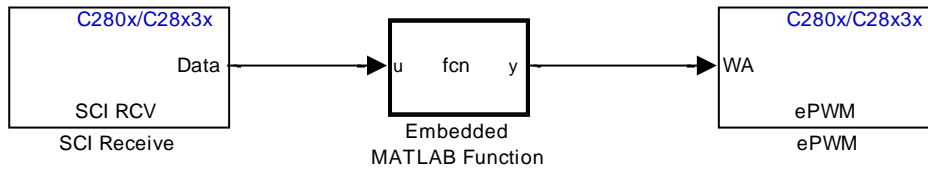
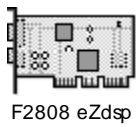
Al circuito PWM Z12A8DDC (Figura 3) se le conectó la alimentación de 24VDC y GND en el conector marcado “+24VDC” y el motor en el conector marcado “motor”, posteriormente se tomaron las señales PWM4, dir4, Gnd (figura 5) y se llevaron hasta el conector marcado “DSP” del circuito de la tarjeta servoamplificadora Z12A8DDC, siguiendo el diagrama esquemático de la figura 4.

La tarjeta servoamplificadora Z12A8DDC no requiere de señales de “enable” ni configuraciones extras, viene lista de fábrica para operar.

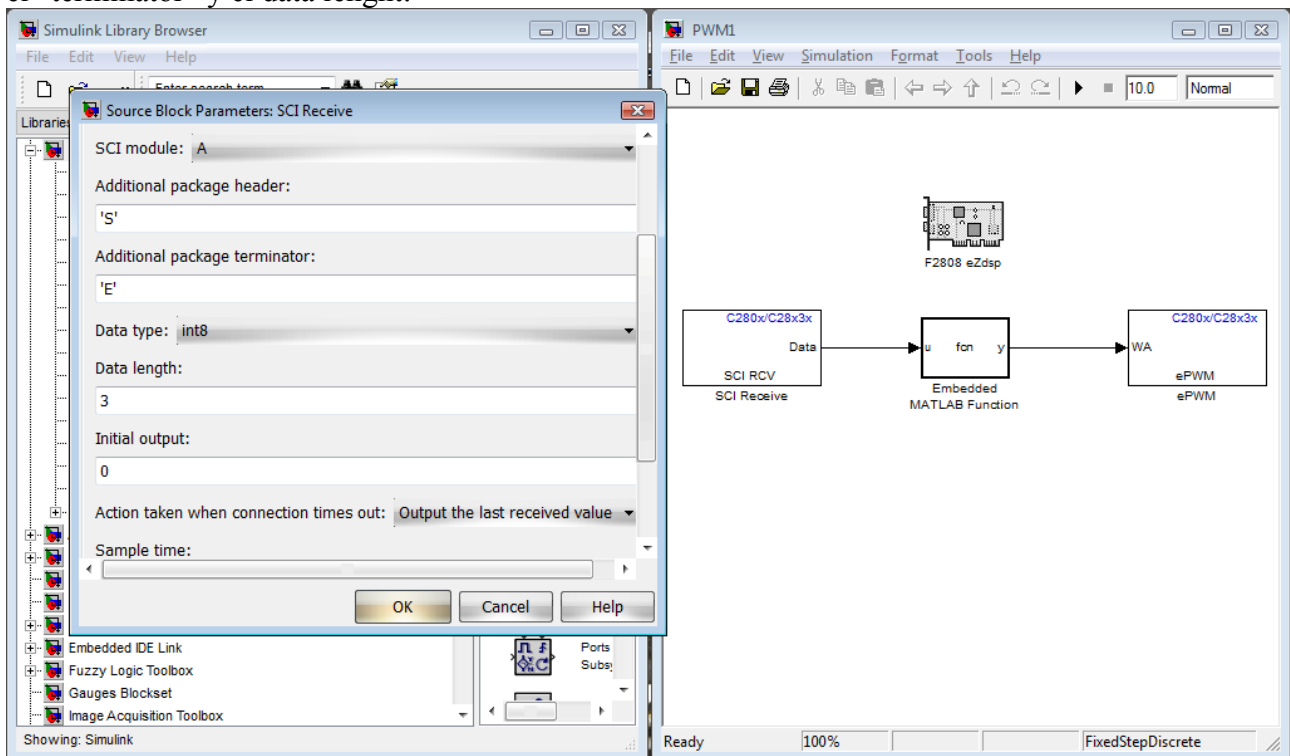
Se hizo un programa sencillo de prueba en el DSP que hace lo siguiente:

Lee el puerto serial SCIA y si detecta que se activa el motor 4, genera una trama de pwm con 50% de duty cycle, a la salida PWM4 (figura5), en el mismo dato leído del puerto serial, también se obtiene la dirección del giro, para lo cual programa un “1” o un “0” en el pin “dir”. El programa de prueba realizado en simulink, es el de la figura 6, junto con el código del bloque “Embedded matlab function”<sup>1</sup>

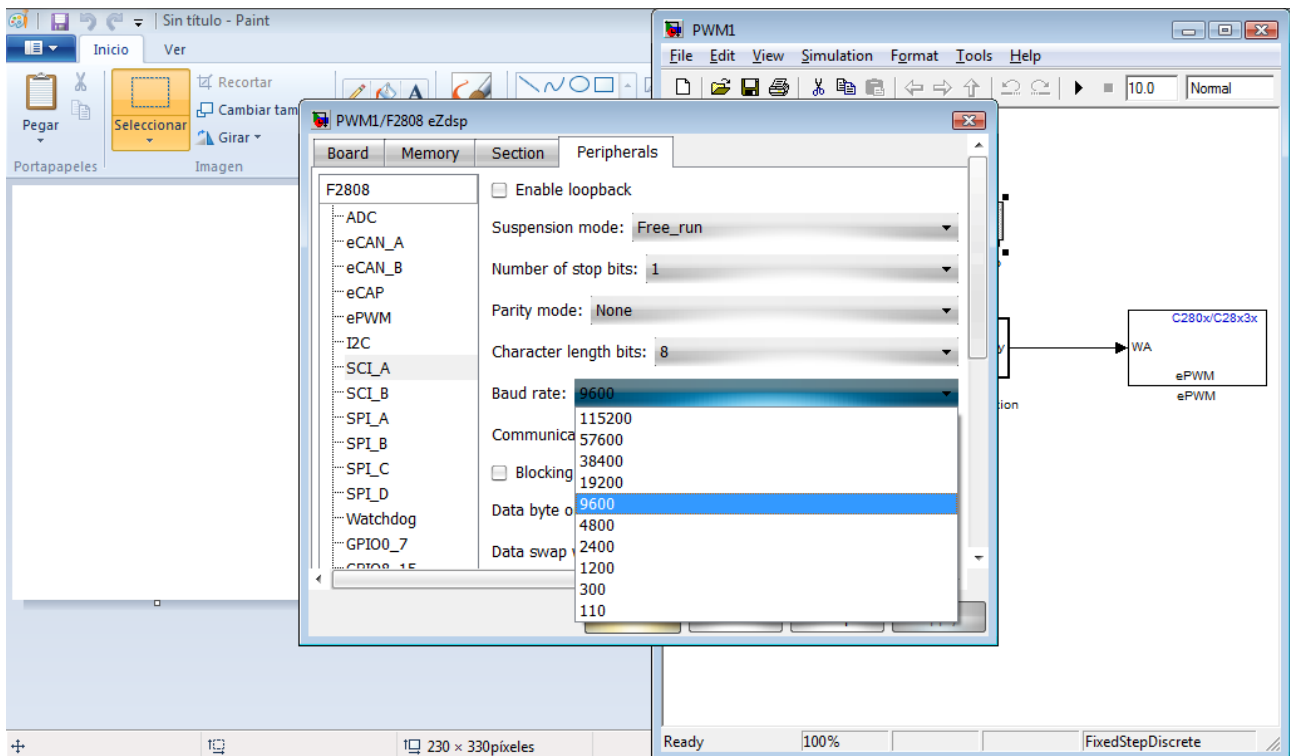
1 Las instrucciones de configuración de matlab y code composer están en el entregable #1



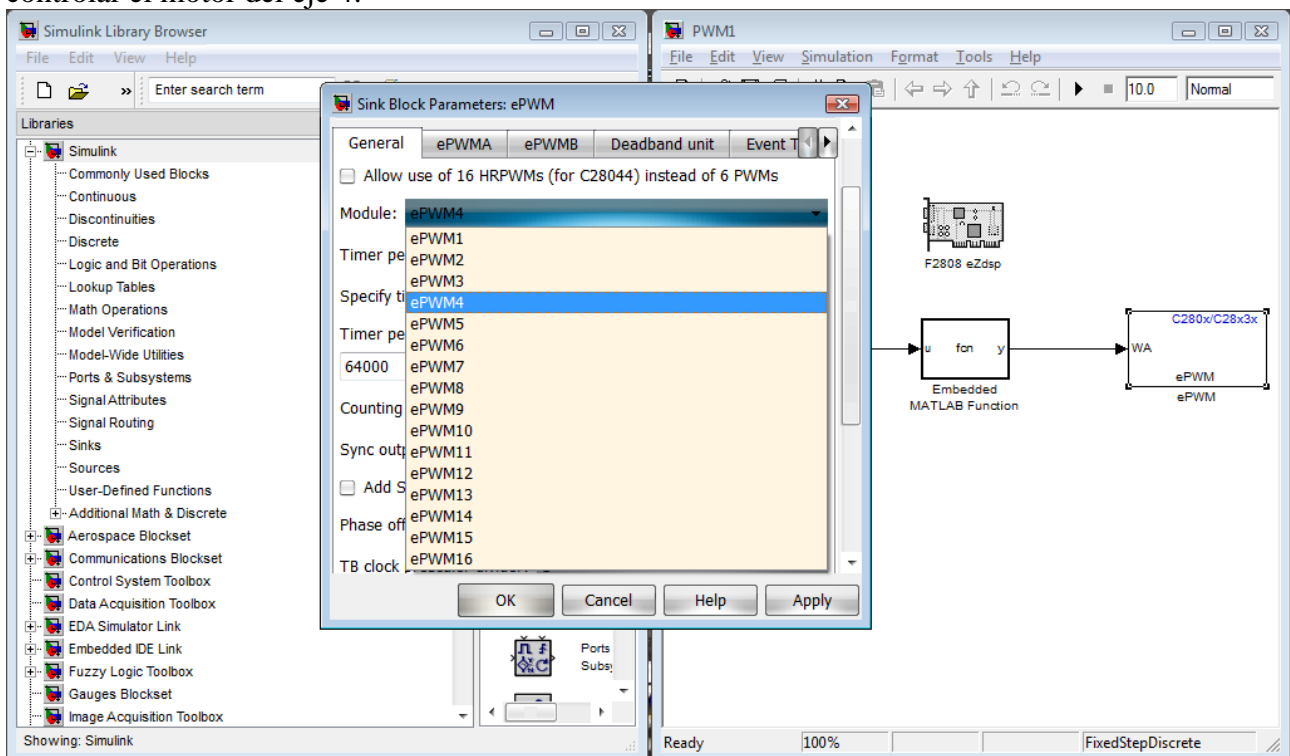
Al hacer doble click sobre el bloque SCI Reciver se puede apreciar la configuración de “header” y el “terminator” y el data length:



Después se hace click sobre el bloque F2808 eZdsp y se configura la velocidad de transmisión de datos:



Luego hacemos doble click sobre el bloque ePWM y configuramos el bloque seleccionando PWM4 ya que corresponde eléctricamente con el pin de PWM4 de la salida del DSP y que se encarga de controlar el motor del eje 4:



Luego hacemos click en la pestaña ePWMA del mismo bloque y colocamos los parámetros tal como se muestran a continuación:

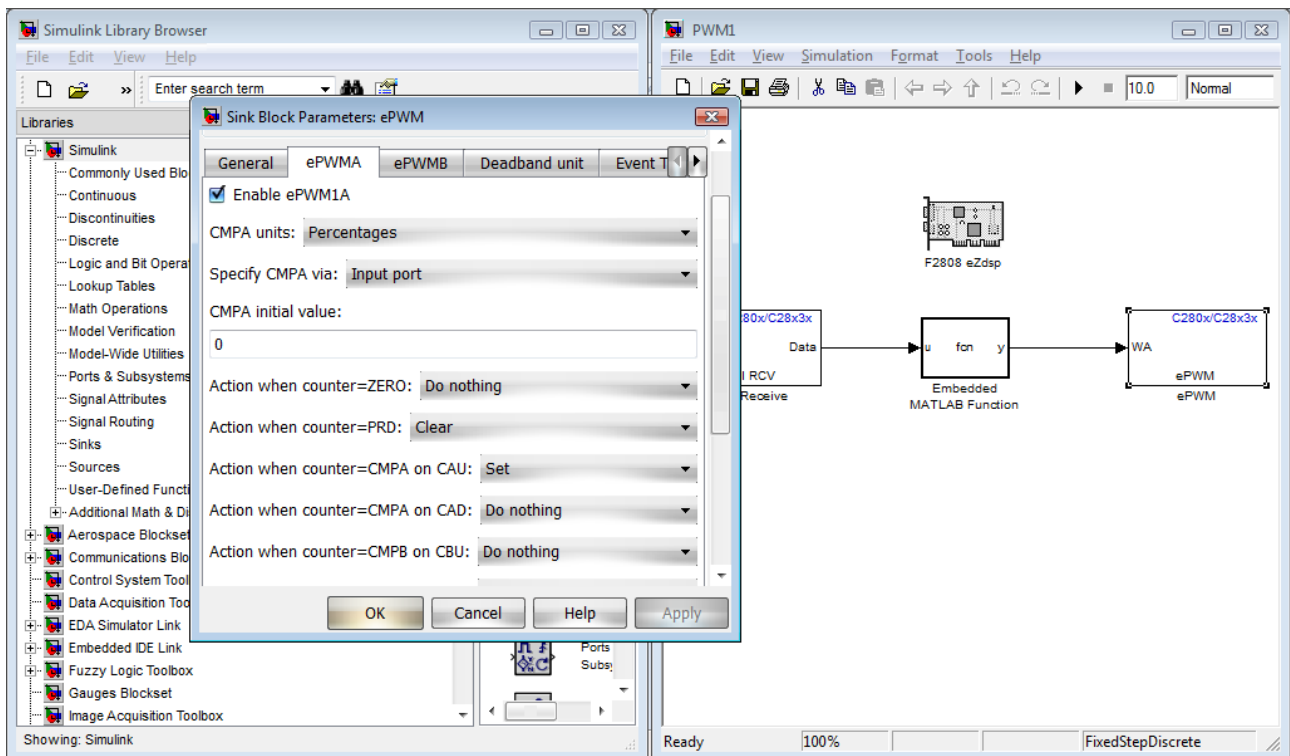


Figura 6.

Finalmente hacemos doble click en el bloque Embedded matlab function y escribimos el siguiente código:

```
function y = fcn(u)
%#eml

% u() es el vector en donde se encuentran los datos
% recibidos por el puerto serial
if (u(2) == '4' & u(4) ~= '0') %u(2) indica cual eje se moverá
y = uint16(50); % 50% del duty cycle
else
y = uint16(0); % 0% del duty cycle
end
```

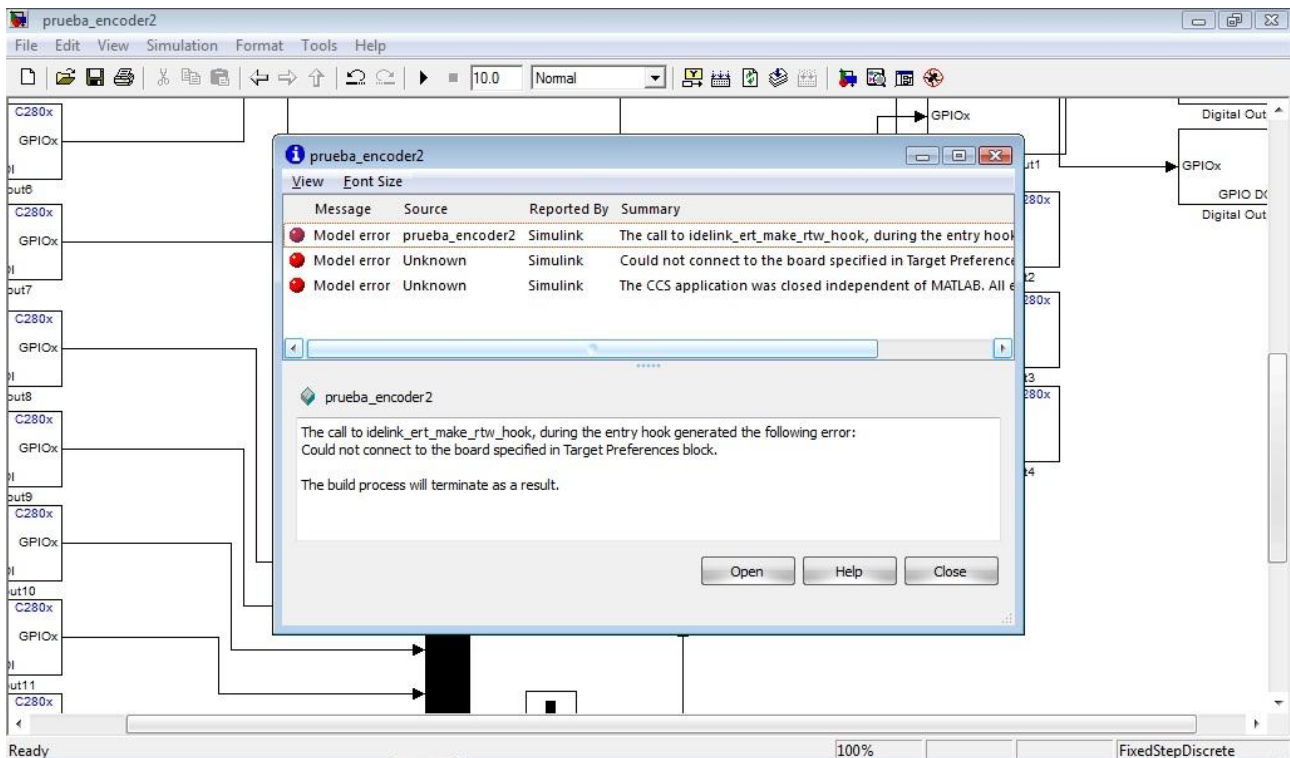
La herramienta de matlab, Embedded coder exige que se especifique el tipo de dato, es por eso que la salida “y” es del tipo uint (unsigned int).

Luego de esto se graba el proyecto y se presiona simultáneamente las teclas Ctrl-B. Esto hace que embedded coder comience a generar el código en C++ y si no encuentra errores, entonces abre el programa Code Composer Studio automáticamente, el cual compila el código generado y lo descarga en la memoria del DSP.

NOTA: Matlab debe abrirse tal como se explicó en el entregable #3, es decir “ejecutar como administrador” y es obligatorio tener el DSP conectado y encendido.

En ocasiones puede salir el siguiente error:





Este error significa que se perdió la conexión entre el programa Code Composer Studio y el DSP, para corregirlo hay que revisar la conexión del puerto USB del DSP y reiniciar matlab y Code Composer, incluso cerrandolo desde la barra de aplicaciones de Windows (parte inferior derecha).

## RESULTADOS

El motor gira bien, con buena potencia, y responden a cambios en la velocidad, modificando el valor del duty cycle en el programa del DSP, también responde bien a cambios en el sentido de giro.

## PROBLEMA ENCONTRADO

Cuando se probó el valor de 0% de duty cycle; el motor giró a maxima velocidad, esto no lo debería hacer, porque con un valor de 0% en el duty cycle, el motor debería estar completamente detenido. Con el osciloscopio se observa la señal del PWM4 en la salida del DSP y se encontró una señal de 5VDC constantes cuando el duty cycle es del 0%, esto no corresponde a ningún error de programación ni falla del DSP, es sólo un criterio que hay que tener en cuenta al momento de diseñar los circuitos, ya que las tarjetas servoamplificadoras fabricadas por FERNEY ALEJANDRO ALVAREZ BENITEZ ya cuentan con un circuito inversor.

## SOLUCIÓN

Se implementa un circuito inversor con un circuito integrado 74HC00N, el cual es TTL compatible, y se conecta a la salida de la señal PWM3 y PWM4 con el fin de invertir la señal y así lograr un nivel de 0VDC cuando el DSP requiere 0% del duty cycle.

El circuito se implementa en un circuito impreso universal siguiendo el diagrama esquemático de la figura 7.

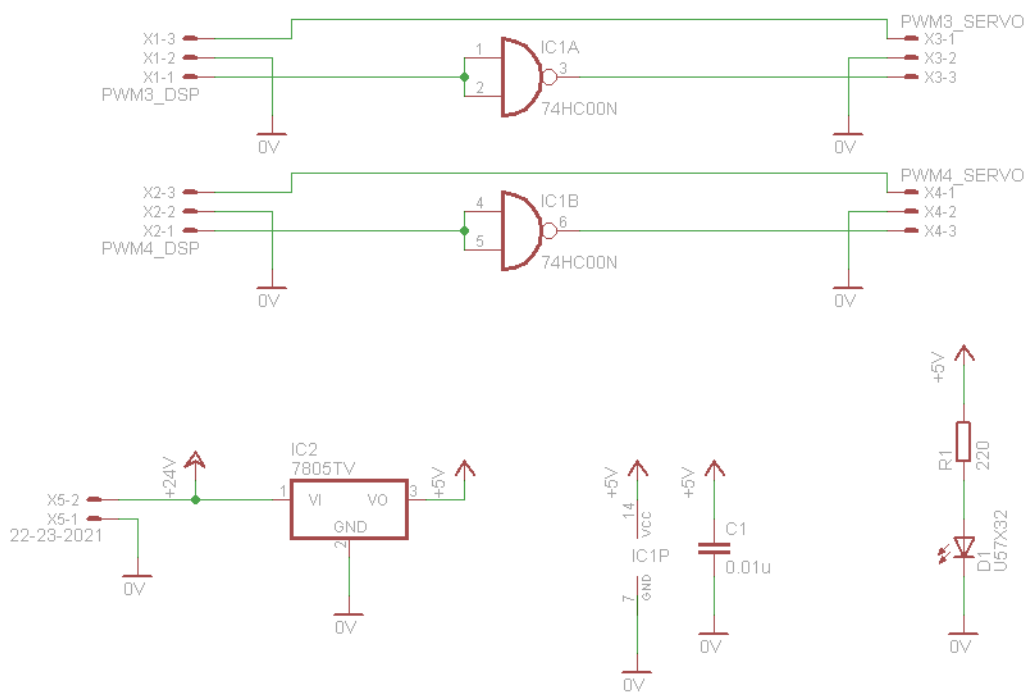


Figura 7

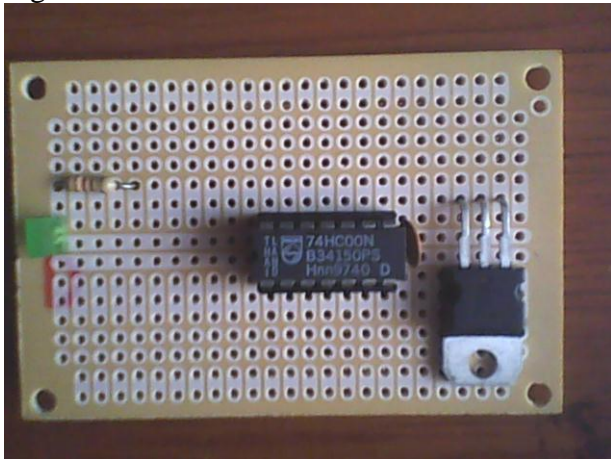


Figura 8 Circuito terminado

## PROBLEMA ENCONTRADO

-Los motores 3 y 4 no responden bien a bajas revoluciones; cuando se configura el DSP para una velocidad equivalente al 5% del duty cycle en el PWM el motor no se mueve, como si le hiciera falta potencia, sin embargo al modificar la velocidad al 8% del duty cycle en el PWM, el motor gira pero a una velocidad que hace difícil el control con el joystick. Los motores 3 y 4 NO son los originales y tienen unos parámetros de torque diferentes a los utilizados en el diseño original, por lo que se trabajaran a 8% de velocidad mientras se gestiona la instalación de los originales.

## SOLUCIÓN

Se configura la velocidad mínima posible del motor 3 en 8% y el motor 4 en 3%, las cuales siguen siendo rápidas pero con un poco de practica se logra obtener un buen control.



## INSTALACIÓN DEL CABLE SERIAL RS232 Y DE VIDEO DESDE EL PODIO HASTA EL SERVIDOR.

En mediciones realizadas, se determinó que hay una distancia aproximada de 30 metros (no directos) entre el podio y el servidor IBM, por lo que la comunicación serial requeriría de un circuito repetidor a los 15 metros. Sin embargo durante pruebas realizadas con la conexión directa entre el DSP y el servidor ubicado a 30 metros, la señal llega sin problemas.

El diagrama esquemático de los cables es el siguiente:

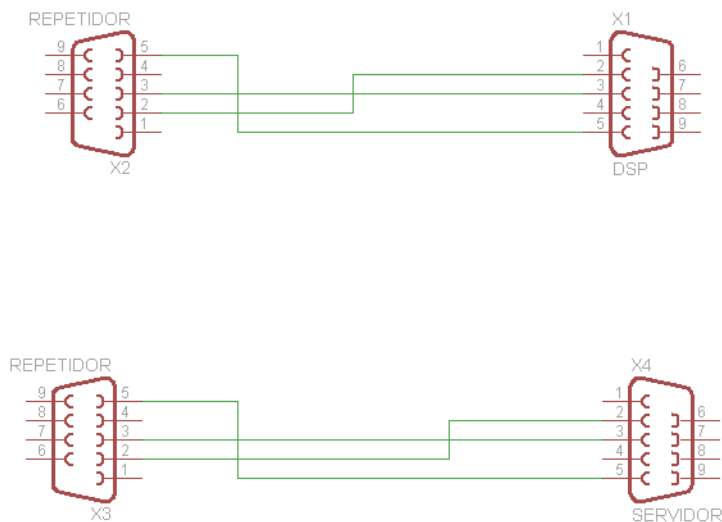
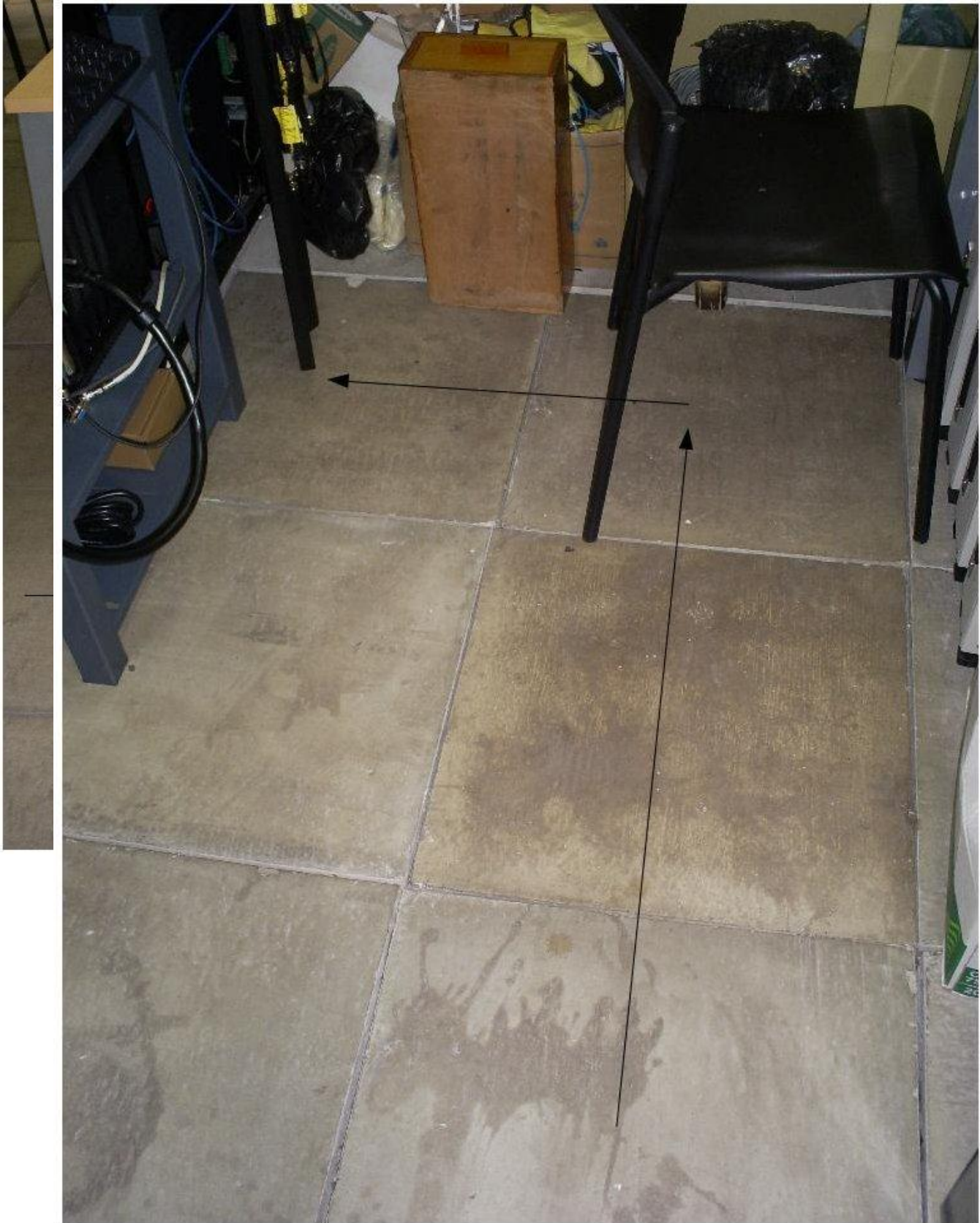


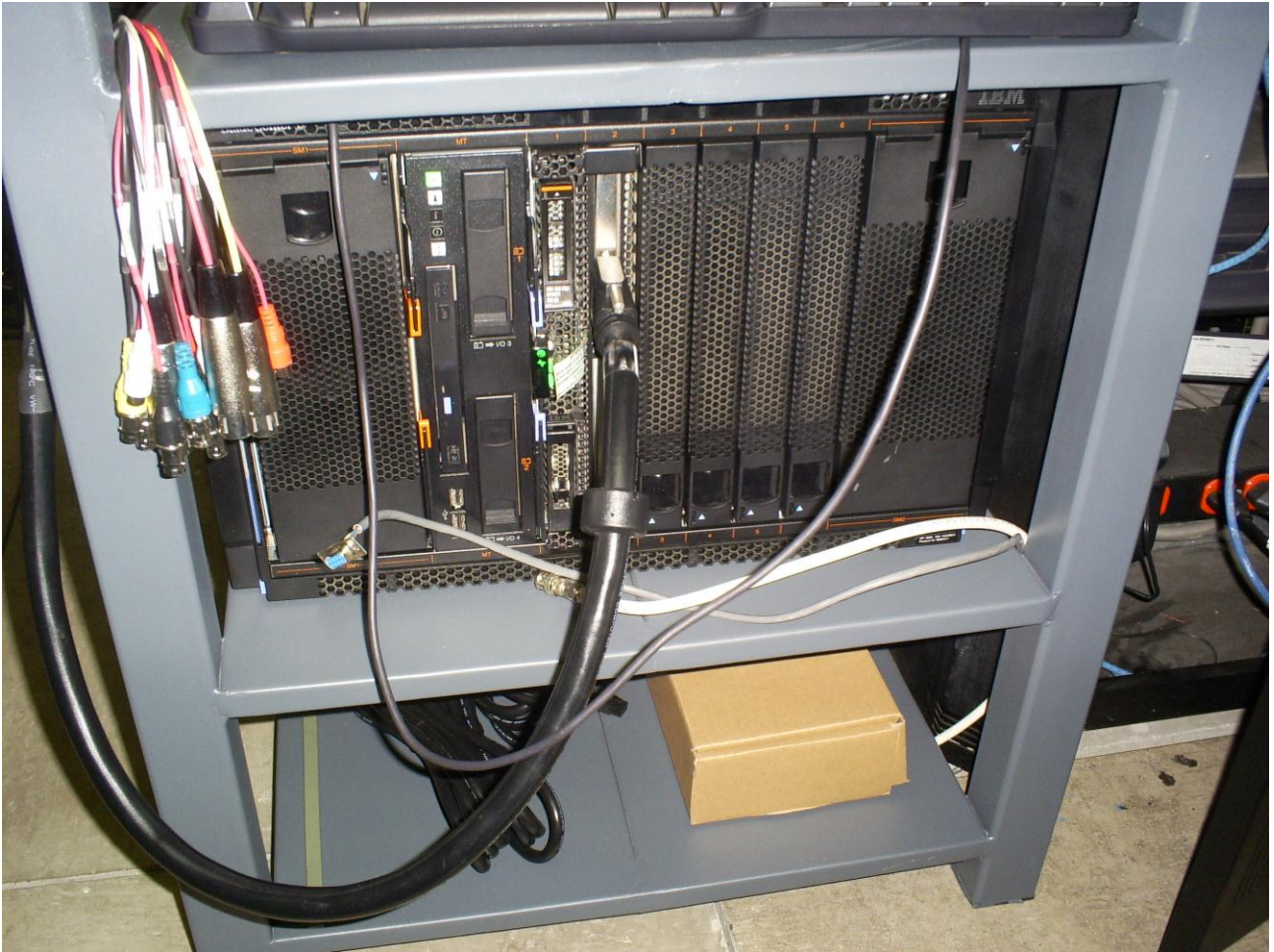
Figura 11

Tanto el cable serial como el de video se instalaron bajo las baldosas del laboratorio de mecatrónica según siguientes indicaciones:

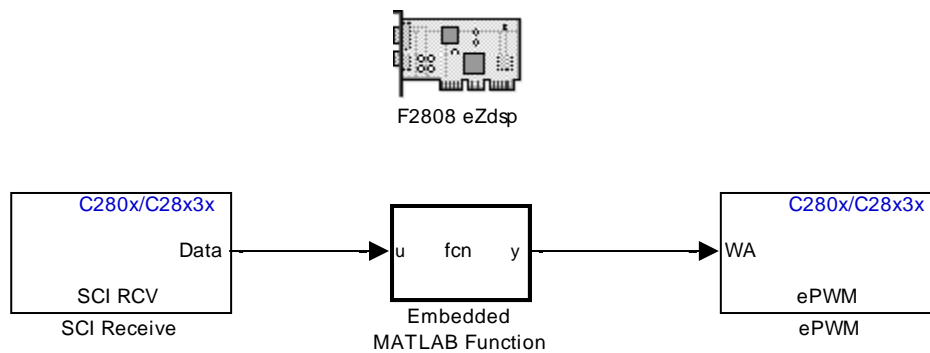






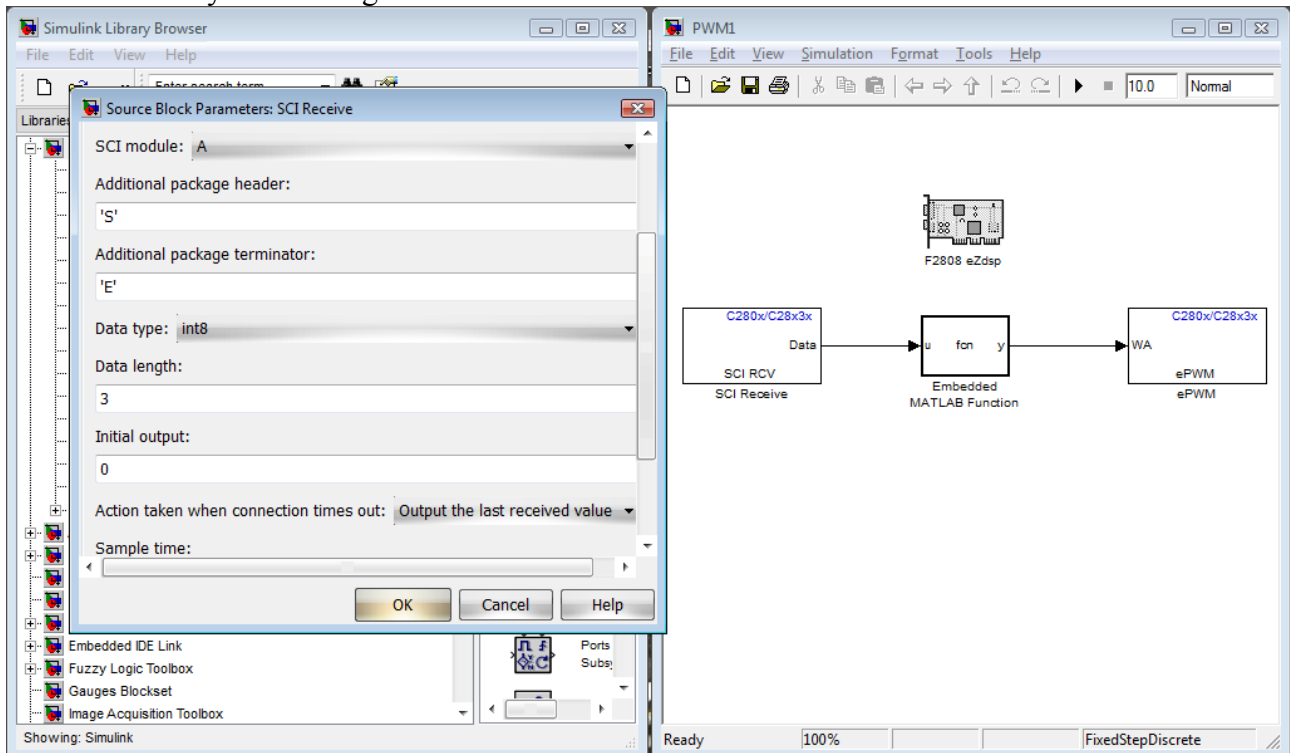


En el DSP se instaló un programa el cual recibe un comando por el puerto serial y hace mover un motor, en este caso el motor #3, el programa de prueba y la configuración requerida es la siguiente:

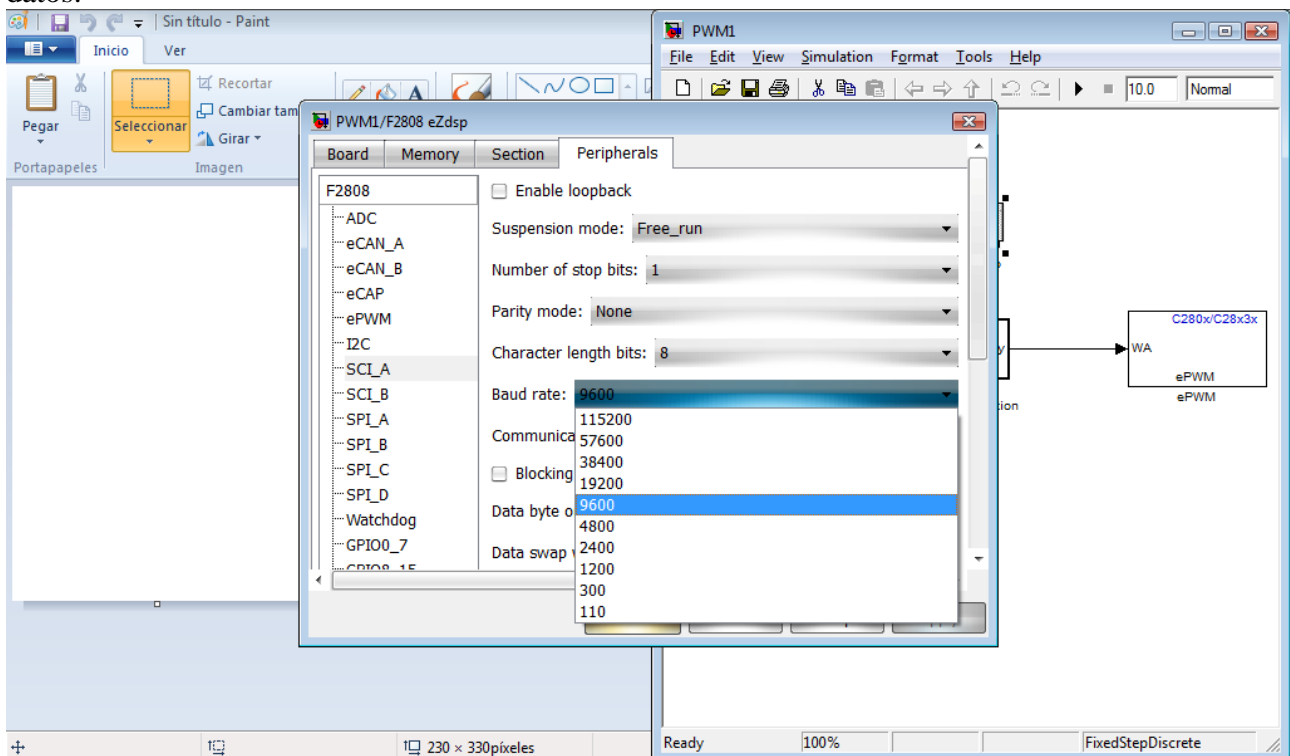


Al hacer doble click sobre el bloque SCI Reciver se puede apreciar la configuración de “header” y

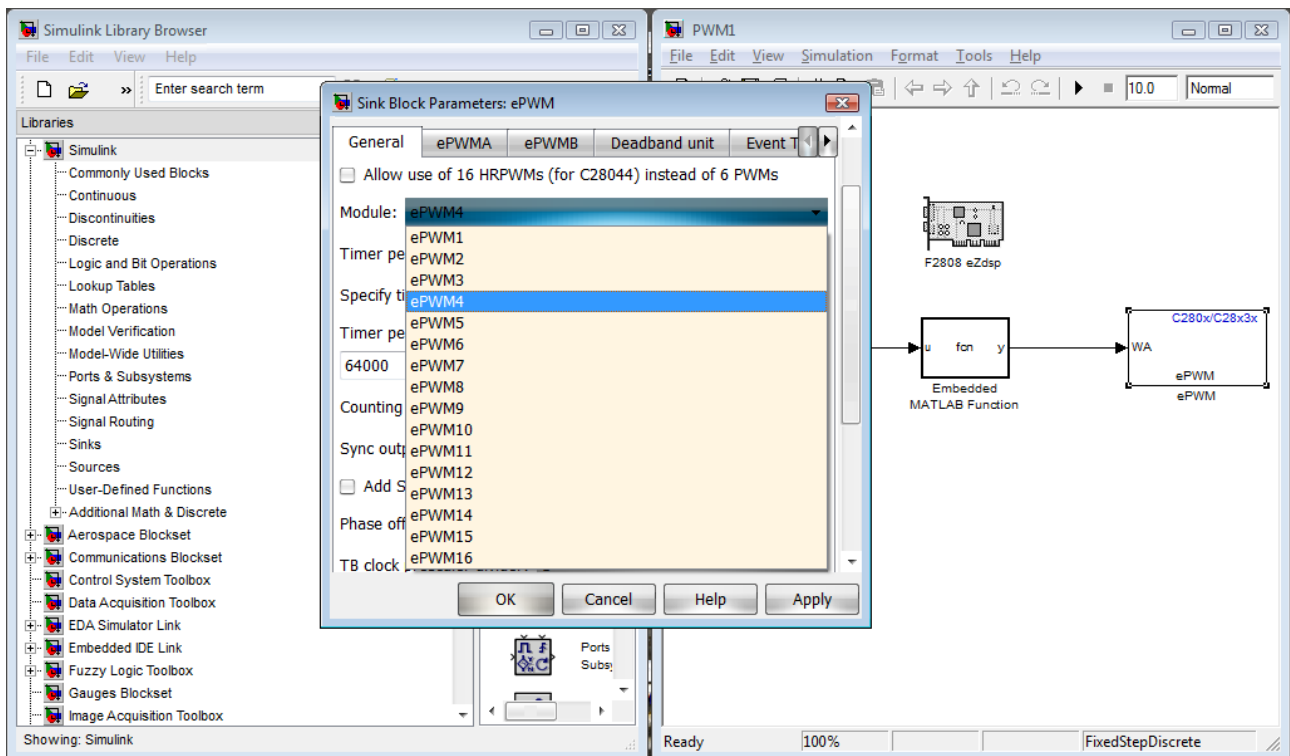
el “terminator” y el data length:



Después se hace click sobre el bloque F2808 eZdsp y se configura la velocidad de transmisión de datos:



Luego hacemos doble click sobre el bloque ePWM y configuramos el bloque seleccionando PWM4 ya que corresponde eléctricamente con el pin de PWM4 de la salida del DSP y que se encarga de controlar el motor del eje 4:



Luego hacemos click en la pestaña ePWMA del mismo bloque y colocamos los parámetros tal como se muestran a continuación:

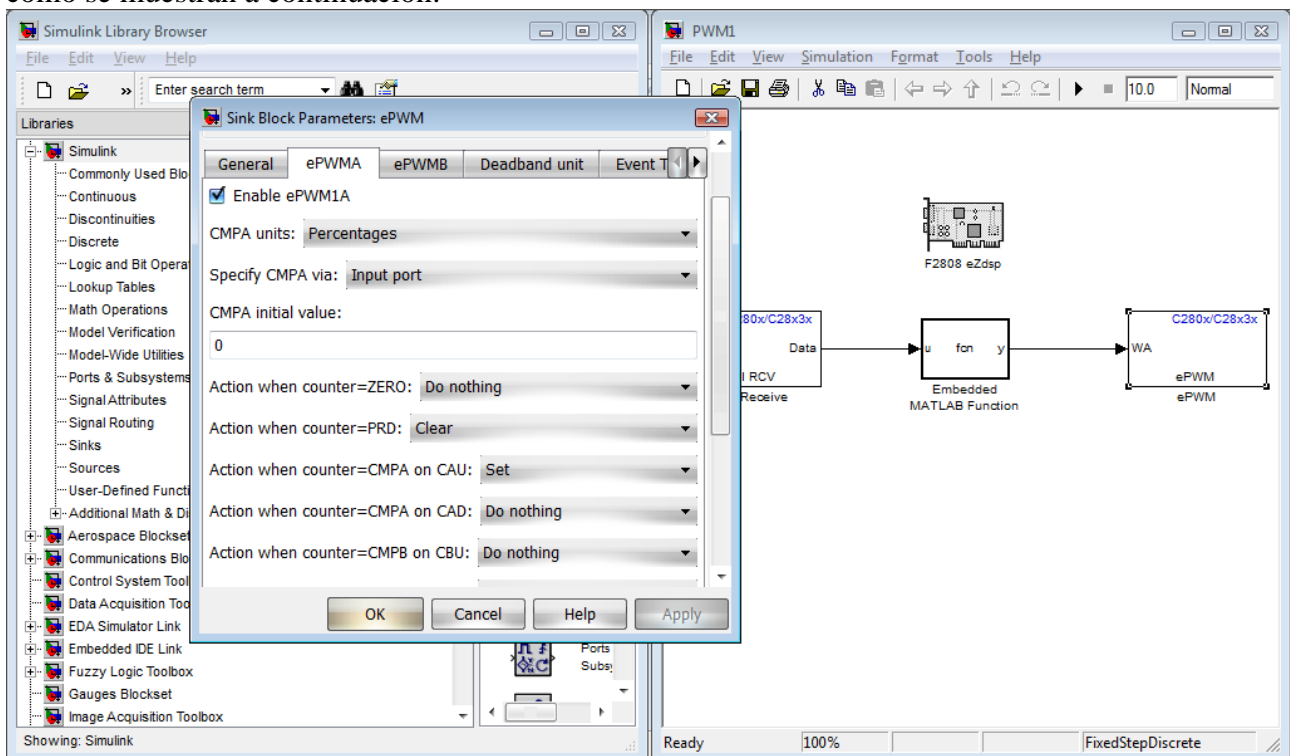


Figura 6.

Finalmente hacemos doble click en el bloque Embedded matlab function y escribimos el siguiente código:

```
function y = fcn(u)
%#eml
```

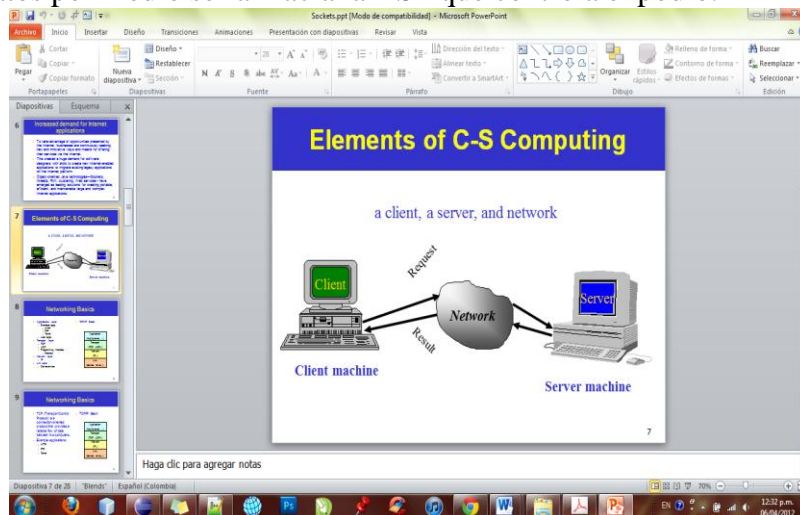
```
% u() es el vector en donde se encuentran los datos
```

```

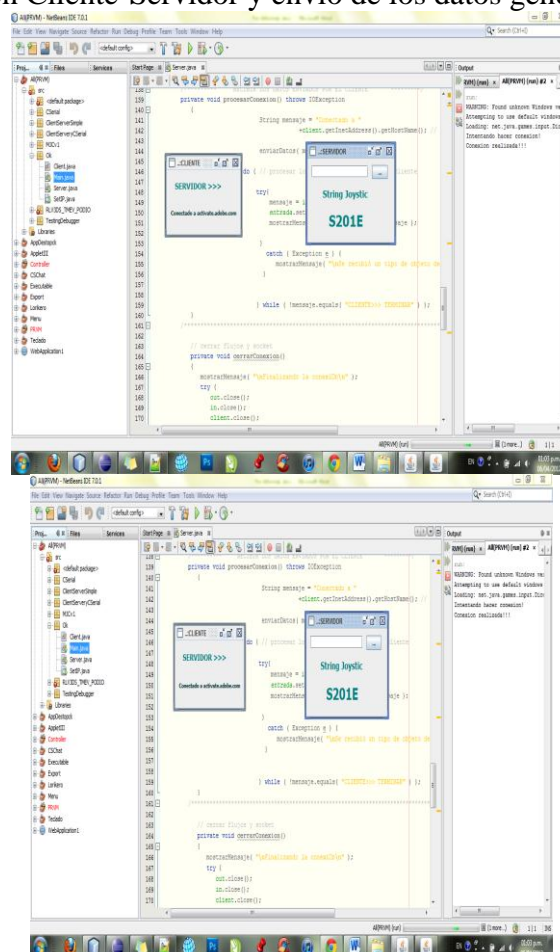
% recibidos por el puerto serial
if (u(2) == '4' & u(4) ~= '0') %u(2) indica cual eje se moverá
y = uint16(50); % 50% del duty cycle
else
y = uint16(0); % 0% del duty cycle
end

```

Envío de los datos generados por el joystick desde el pc IBM (Cliente) hacia el Servidor y posterior envío de dichos datos por medio serial hacia la DSP que controla el podio.



Se logró establecer conexión Cliente-Servidor y envío de los datos generados por el joystick.

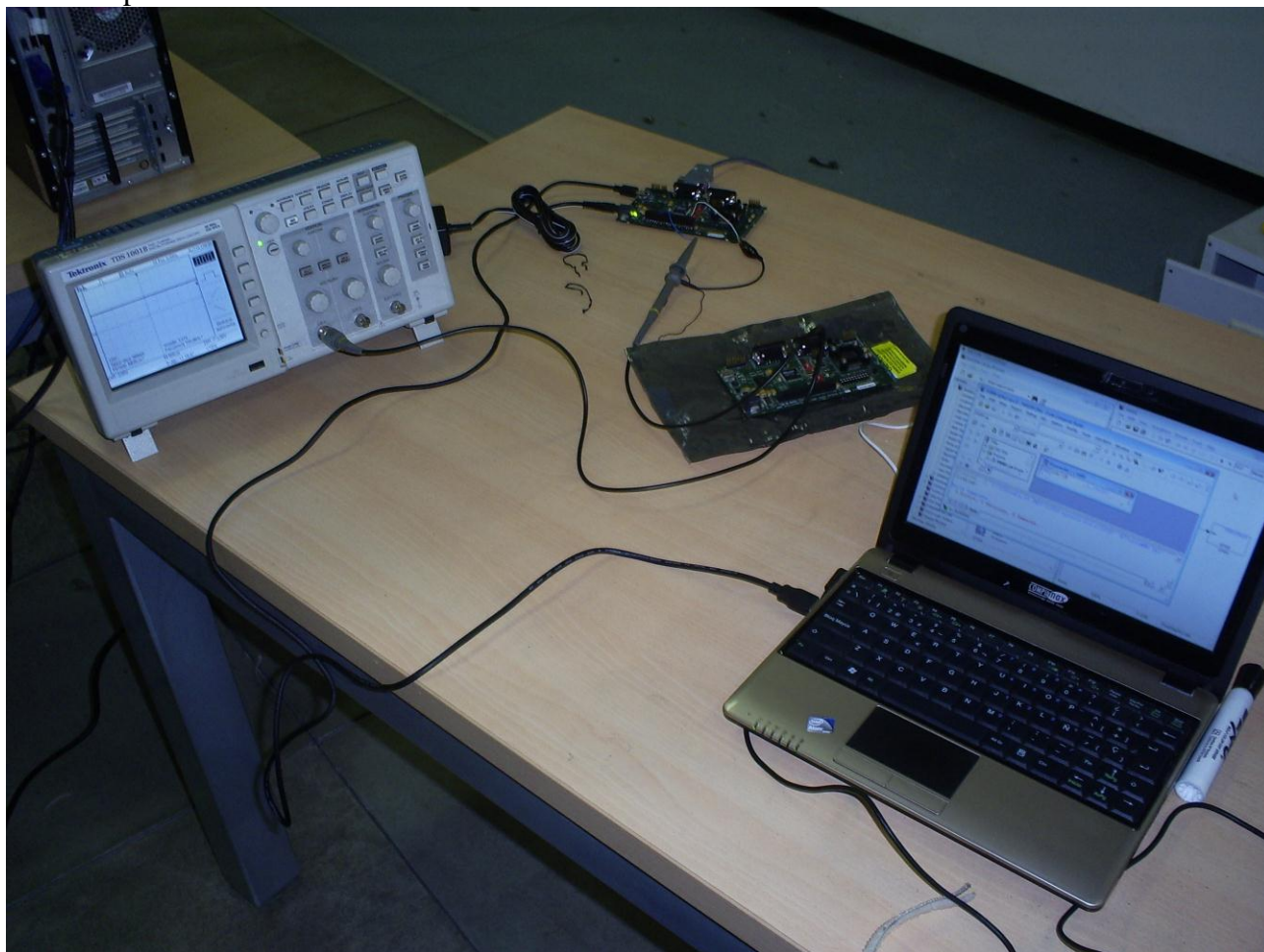


Las imágenes anteriores muestran la conexión y el posterior recibimiento del servidor de los datos enviados por el cliente.

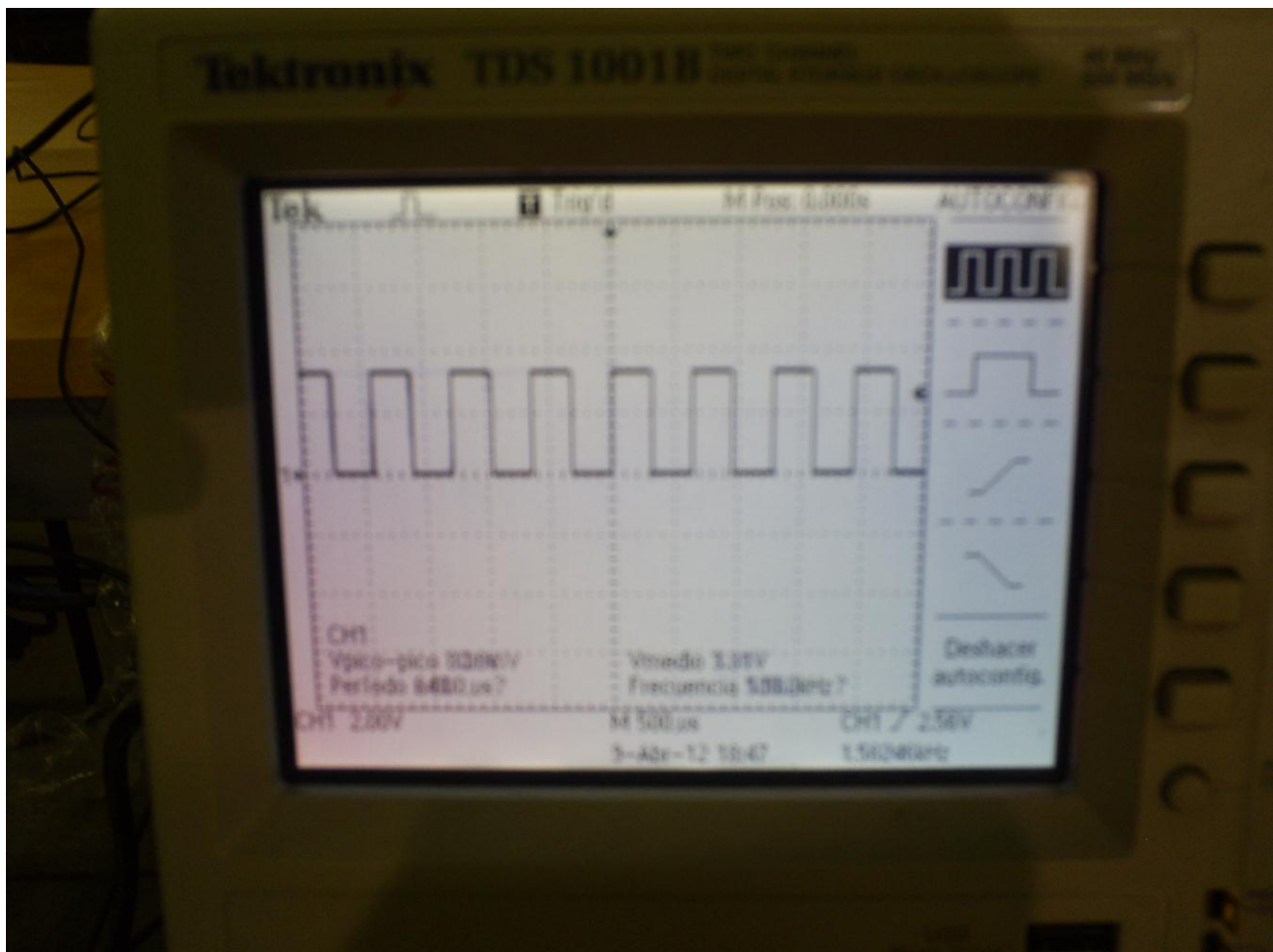
Posterior a esto se intentó enviar los datos hacia la DSP y se logró hacer comunicación vía puerto serial entre el servidor IBM y el repetidor, cuyos resultados fueron observados por medio de un



osciloscopio:



Prueba realizada en la conexión serial a 15 metros del servidor.



En esta imagen se aprecia una señal PWM a la salida del DSP, cada vez se que ejercia una orden desde el Joystick conectado en la estación cliente y que pasaba al servidor vía internet.

