



# INSTITUTO POLITÉCNICO NACIONAL CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

No. 112

Serie: VERDE

Fecha: Agosto 2006

## Guía de uso de la plataforma Gumstix Basix 200-Bt

Ing. Arturo Rojo Ruiz\*  
M en C Amadeo José Arguelles Cruz†

### RESUMEN

Este documento presenta una guía para el uso del “Gumstix Basix 200-BT”, tanto su configuración como una breve iniciación en su forma de programación.

**Palabras clave:** Gumstix, Linux embebido, Sistemas dedicados, Usbnet, Bluetooth.

\*Alumno de la Maestría en Ciencias en Ingeniería de Cómputo del C.I.C., #Teléfono 0445517029452, [arturo.rojo@gmail.com](mailto:arturo.rojo@gmail.com)

†Profesor-Investigador de la Maestría en Ciencias en Ingeniería de Cómputo del C.I.C., #Teléfono 5729 6000 ext.56519

## **ADVERTENCIA**

“Este reporte contiene información desarrollada por el Centro de Investigación en Computación del Instituto Politécnico Nacional a partir de datos y documentos con derechos de propiedad y por lo tanto su uso queda restringido a las aplicaciones que explícitamente se convenga.

La aplicación no convenida exime al Centro de su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte podrá obtenerse recurriendo a la Unidad de Publicaciones y Reportes Técnicos del centro de Investigación en Computación del I.P.N. Av. Juan de Dios Bátiz s/n, teléfono 57-29-60-00 ext. 56500, 56608 y 56610”.

# ÍNDICE

|  |    |
|--|----|
| <b>1. INTRODUCCIÓN</b> .....   | 1  |
| Figura 1.1 Gumstix Basic 200-bt.....                                     | 1  |
| <b>2. REQUERIMIENTOS</b> .....   | 1  |
| <b>3. CONFIGURACIÓN</b> .....  | 2  |
| <b>3.1 Usbnet</b> .....  | 2  |
| Figura 3.1 Acceso al gumstix usando ssh.....                             | 3  |
| Figura 3.2 Servidor Web del gumstix.....                                 | 3  |
| <b>3.2 Bluetooth</b> .....   | 4  |
| Figura 3.3 Resultado del comando ps --aux   grep pand   grep --v grep    | 6  |
| Figura 3.4 Seleccionar Gumstix(0) en la búsqueda.....                    | 6  |
| Figura 3.5 Seleccionar Punto de acceso a redes en lista de servicios.... | 6  |
| Figura 3.6 Propiedades TCP/IP de la Conexión de Área Local 2.....        | 7  |
| Figura 3.7 Conectando una red PAN vía Bluetooth.....                     | 7  |
| <b>4. PROGRAMACIÓN</b> .....   | 8  |
| <b>4.1 Buildroot</b> .....   | 8  |
| <b>4.2 Ejemplos</b> .....  | 9  |
| <b>4.2.1 Hello.c</b> .....   | 9  |
| Figura 4.1 Ejecución del programa hello.....                             | 10 |
| <b>4.2.2 Socketc.c</b> .....   | 10 |
| Figura 4.2 Form1 de sockets.....   | 12 |
| Figura 4.3 Ejecución de sockets.....                                     | 13 |
| Figura 4.4 Ejecución de socketc.....                                     | 13 |
| <b>5. BIBLIOGRAFÍA</b> .....   | 14 |
| <b>6. ANEXOS</b> .....   | 15 |
| Tabla 6.1 Conector Hirose 60 en Gumstix 200-bt.....                      | 15 |

## 1. INTRODUCCIÓN

El Gumstix Basix 200-bt (Figura 1.1) es una plataforma que sirve para implementar sistemas dedicados.

Contiene un procesador Intel Xscale PXA255, un reloj de 200Mhz, 64Mb de RAM y 4Mb de memoria *flash*, lector de memoria MMC (MultiMedia Card), Bluetooth y trae precargado Linux. Incluye un conector Hirose de 60 terminales, en el cual se encuentran terminales de E/S de propósito general así como puertos seriales, USB, LCD, etc. Para más información ver Anexo A.

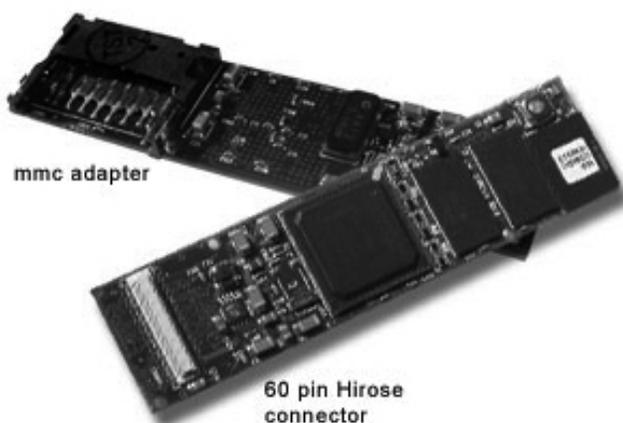


Figura 1.1 Gumstix Basic 200-bt

El objetivo de esta guía es mostrar como llevar a cabo su configuración, como utilizar las herramientas necesarias para programarlo, y una breve introducción a la forma de programación.

## 2. REQUERIMIENTOS

Hardware:

- Una PC con Linux preferentemente Mandriva con KDE.
- Un Gumstix Basix 200-bt.
- Una tarjeta de expansión Thumbstix-gs

Software:

- SSH
- Servidor DHCP
- Servidor Web (Apache)
- GCC

### 3. CONFIGURACIÓN

#### 3.1 Usbnet (USB Networking)

Usbnet es un protocolo para operar a una red TCP/IP sobre USB. Esto permite ver al gumstix como un host en una red. Para realizar la configuración, es necesario que se proceda como se indica a continuación:

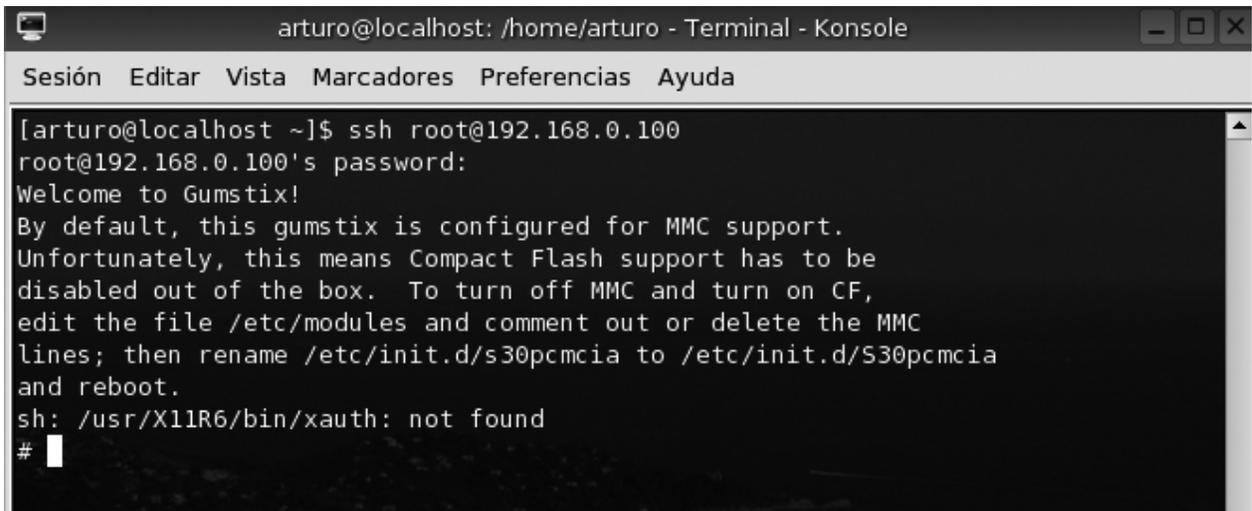
1. Primero conecte el gumstix con la tarjeta de expansión thumbstix por medio del conector Hirose 60.
2. Conecte el gumstix a un puerto USB de la PC que contiene a Linux.
3. Abra una consola y en la línea de comandos teclee el comando su (súper usuario), y después su clave de acceso de (súper usuario).
4. Posteriormente teclee ifconfig -a para ver si muestra usb0 (esto puede tomar algunos minutos después de conectado el gumstix).
5. Teclee ifconfig usb0 192.168.0.1 esto la asigna dicha dirección a la interfaz usb0.
6. Teclee route add -host 255.255.255.255 dev usb0 esto es necesario para poder configurar el servidor dhcp en la interfaz usb0.
7. Edite /etc/dhcpd.conf y modifíquelo para que se vea de la siguiente manera:

```
ddns-update-style ad-hoc;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
option domain-name-servers 192.168.0.1;
max-lease-time 120;
default-lease-time 120;

subnet 192.168.0.0 netmask 255.255.255.0 { range 192.168.0.10 192.168.0.100;}
```

8. Ahora solo se tiene que activar el servidor dhcp tecleando /usr/sbin/dhcpd -d -f
9. Cierre esa consola, abra otra y teclee ssh [root@192.168.0.100](http://root@192.168.0.100) , y cuando le pida el password teclee gumstix.

Ya esta dentro del gumstix (figura 3.1).

A terminal window titled 'arturo@localhost: /home/arturo - Terminal - Konsole'. The window has a menu bar with 'Sesión', 'Editar', 'Vista', 'Marcadores', 'Preferencias', and 'Ayuda'. The terminal content shows a user running 'ssh root@192.168.0.100'. The remote host prompts for a password, which is entered. The user is then greeted with 'Welcome to Gumstix!' and a message about MMC and CF support. The user then enters 'sh: /usr/X11R6/bin/xauth: not found' and the prompt changes to '#'.

```
arturo@localhost: /home/arturo - Terminal - Konsole
Sesión  Editar  Vista  Marcadores  Preferencias  Ayuda
[arturo@localhost ~]$ ssh root@192.168.0.100
root@192.168.0.100's password:
Welcome to Gumstix!
By default, this gumstix is configured for MMC support.
Unfortunately, this means Compact Flash support has to be
disabled out of the box.  To turn off MMC and turn on CF,
edit the file /etc/modules and comment out or delete the MMC
lines; then rename /etc/init.d/s30pcmcia to /etc/init.d/S30pcmcia
and reboot.
sh: /usr/X11R6/bin/xauth: not found
#
```

Figura 3.1 Acceso al gumstix usando ssh

Otra forma de probar que ya está configurado su gumstix, es en algún explorador buscar la dirección <http://192.168.0.100> (figura 3.2)



Figura 3.2 Servidor Web del gumstix

En caso de que desee asignarle una IP fija haga lo siguiente.

En el gumstix:

1. Edite /etc/network/interfaces

```
#iface usb0 inet dhcp
iface usb0 inet static
    address 192.168.0.100
    netmask 255.25.255.0
```

### 3.2 Bluetooth

Hay dos formas de configurar el bluetooth, la primera es que el gumstix este como servidor y la PC con Linux como cliente y la segunda es configurar el gumstix como cliente y habilitar una PC con Windows XP como servidor.

Para el primer caso, se tiene que hacer lo siguiente:

En el gumstix:

1. Edite el archivo ubicado en la ruta `/etc/network/interfaces`

```
#iface bnep0 inet dhcp
iface bnep0 inet static
    address 192.168.69.1
    netmask 255.25.255.0
```

2. Edite el archivo ubicado en la ruta `/etc/init.d/S30bluetooth`

```
#PAND_ENABLE=false
PAND_ENABLE=true
```

3. Abrir una consola y teclear el comando `/etc/init.d/S30bluetooth stop`
4. Teclear el comando `/etc/init.d/S30bluetooth start`
5. Para revisar que ya esta funcionando el Bluetooth, es necesario teclear el comando `ps --aux | grep pand | grep -v grep`

En la PC con Linux:

1. Abra una consola y en la línea de comandos teclee el comando `su` (súper usuario), y después su clave de acceso de (súper usuario).
2. Teclear el comando `pand --listen --role NAP`
3. Teclee `ifconfig bnep0 192.168.0.2 netmask 255.255.25.0`
4. Teclee `ifconfig bnep0 up`
5. Para probar la conexión teclee `ping 192.168.69.1`

Ya esta configurado el bluetooth.

Algo importante de mencionar es que se puede acceder al gumstix vía ssh por bluetooth, lo cual es importante si se desea programar el gumstix a distancia, esto se hace de la siguiente manera:

Abra una consola y teclee ssh [root@192.168.69.1](ssh://root@192.168.69.1) y el password gumstix

Ahora bien si queremos trabajar con un servidor de Windows XP se debe de hacer lo siguiente:

En el gumstix:

1. Edite el archivo `/etc/network/interfaces`

```
#iface bnep0 inet dhcp
iface bnep0 inet static
    address 192.168.69.1
    netmask 255.25.255.0
```

2. Edite el archivo `/etc/init.d/S30bluetooth`

```
#PAND_ENABLE=false
PAND_ENABLE=true
```

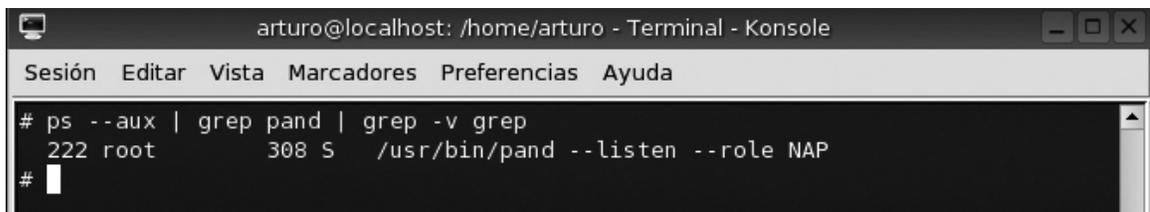
3. Edite el archivo `/etc/bluetooth/hcid.conf`

```
#class 0x820116
class 0x020100
```

4. Edite el archivo `/etc/default/bluetooth:`

```
#PAND_OPTIONS="role PANU --search --persist"
PAND_OPTIONS="--listen --role NAP"
```

5. Teclee `/etc/init.d/S30bluetooth stop`
6. Teclee `/etc/init.d/S30bluetooth start`
7. Para revisar que ya esta funcionando el Bluetooth teclee `ps --aux | grep pand | grep -v grep` (ver figura 3.3)



```
arturo@localhost: /home/arturo - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
# ps --aux | grep pand | grep -v grep
222 root 308 S /usr/bin/pand --listen --role NAP
#
```

Figura 3.3 Resultado del comando `ps --aux | grep pand | grep --v grep`

En la PC con Windows XP:

1. Buscar dispositivos bluetooth al alcance en modo personalizado y seleccionar Gumstix(0) (figura 3.4).

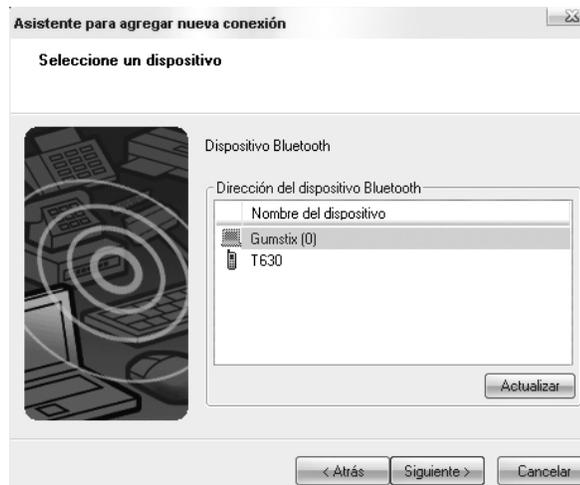


Figura 3.4 Seleccionar Gumstix(0) en la búsqueda.

2. En la lista de servicios seleccionar punto de acceso a redes (figura 3.5) y en el nombre de la conexión ponerle Gumstix Network.



Figura 3.5 Seleccionar Punto de acceso a redes en lista de servicios

3. En conexiones de red, abrir las propiedades de “Conexión de Área Local 2” (dependiendo del adaptador de bluetooth en número de la conexión puede variar) y modificar las propiedades TCP/IP colocándole la IP 192.168.69.2 y en máscara de subred colocarle 255.255.255.0, los demás parámetros se dejan en blanco (figura 3.6).

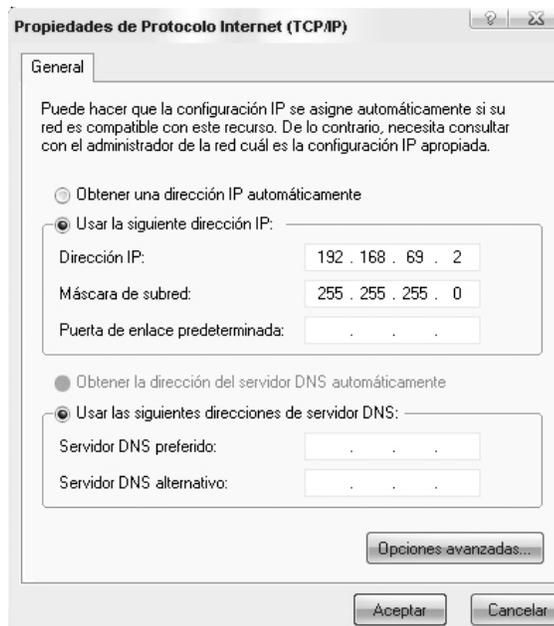


Figura 3.6 Propiedades TCP/IP de la Conexión de Área Local 2

#### 4. Dale conectar en la conexión Gumstix Network

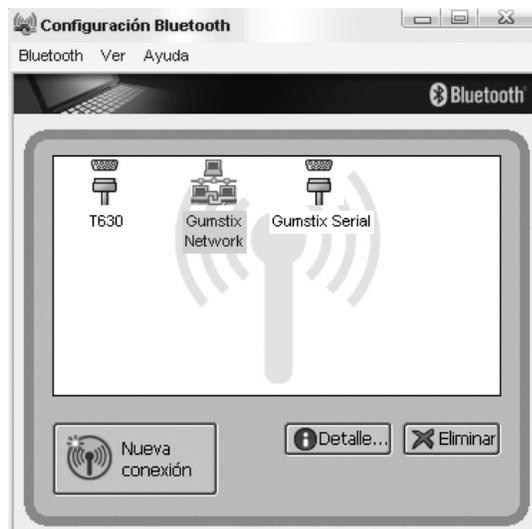


Figura 3.7 Conectando una red PAN vía Bluetooth

#### 5. Probar la conexión haciendo un ping a la dirección 192.168.69.1

## 4. PROGRAMACIÓN

### 4.1 Buildroot

Para escribir un programa para el gumstix se necesita un compilador, el cual se instala en otro sistema Linux, es importante tener espacio suficiente ya que éste requiere cerca de 2.2gb.

Buildroot es una herramienta que contiene todos los programas y funciones de biblioteca necesarios para poder crear programas con los cuales pueda operar la tarjeta gumstix.

Lo primero que se debe de hacer es bajar e instalar un cliente de *Subversion* para poder acceder al repositorio de gumstix y bajar posteriormente el buildroot, este se puede obtener de la siguiente pagina en Internet <http://subversion.tigris.org>.

Una vez que ya esta instalado *Subversion* procedemos a bajar e instalar el buildroot de la siguiente manera:

6. Abra una consola y en la línea de comandos teclee el comando su (súper usuario), y después su clave de acceso de (súper usuario).
1. Teclee `svn co -r 773 http://svn.gumstix.com/gumstix-buildroot/trunk gumstix-buildroot` , donde desea que se baje el buildroot.
2. Teclee `cd gumstix-buildroot`
3. Teclee el comando `make`

Con esto se instalan todas las herramientas para trabajar, pero aún no se puede usar el compilador desde cualquier carpeta, para lograr esto se tiene que realizar lo siguiente:

1. Abra una consola y en la línea de comandos teclee el comando su (súper usuario), y después su clave de acceso de (súper usuario).
2. Teclee `cd /bin`
3. Teclee `ln -s /home/erver/programas/gumstix-buildroot/build_arm_nofpu/staging_dir/bin/arm-linux-gcc arm-linux-gcc`

Con esto se da de alta el compilador arm-linux-gcc en todo el sistema y puede ser llamado desde cualquier lugar.

## 4.2 Ejemplos

### 4.2.1 Hello.c

Que mejor para probar que todo este bien instalado que hacer un pequeño programa, para esto haga lo siguiente:

1. Cree un archivo llamado hello.c que contenga lo siguiente.

```
#include <stdio.h>
void main()
{
    printf("Hello, small world!\n");
}
```

2. Teclee `arm-linux-gcc hello.c -o hello` para compilarlo

Ya tenemos un ejecutable que se puede ejecutar en el gumstix, no lo pruebe en la PC donde lo compilo porque no va a servir.

Ahora solo falta pasarle el ejecutable al gumstix, y esto se puede hacer de dos maneras, una es usando una MMC y la otra es utilizando un servidor Web en la PC con Linux.

Para usar la MMC introduzca ahí el programa usando algún lector de tarjetas en la PC. Después introduzca la MMC en el gumstix, tenga cuidado de que se encuentre apagado, ya que para que la reconozca tiene que encenderse con la MMC puesta. Luego conéctese al gumstix por ssh y teclee `mount /mnt/mmc` y listo, ya puede acceder a la MMC por medio de la carpeta `/mnt/mmc`.

Si quiere pasarle la información por medio de un servidor Web (Apache) haga lo siguiente:

1. Abrir una consola y presionar su, y teclear su password de root.
2. Teclee `cd /var/www/html`
3. Teclee `mkdir programas`
4. Teclee `chmod 777 programas`
5. Copie el archivo ejecutable hello a `/var/www/html/programas`
6. Entre al gumstix usando ssh
7. Teclee `wget http://192.168.0.1/programas/hello`
8. Teclee `./hello` (puede ser que tenga que modificar los permisos usando `chmod`)

Se espera obtener un resultado como se muestra en la figura 4.1, si marca un error al generarse la segmentación (segmentation fault), significa que la versión del buildroot es errónea y se tiene que bajar otra, para mayor información revise la página en Internet [www.gumstix.org/tikiwiki/tiki-index.php](http://www.gumstix.org/tikiwiki/tiki-index.php)

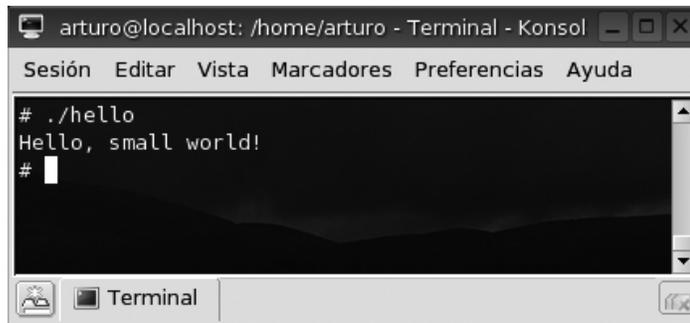


Figura 4.1 Ejecución del programa hello

## 4.2.2 Socketc.c

Este programa se compone de dos partes: una en el gumstix que es el cliente y otra en una PC con Windows XP que esta hecho en C++ Builder 5.

Estando en Linux, debe crear un programa llamado socketc.c que tenga lo siguiente:

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
/* netbd.h es necesitada por la estructura hostent */

#define PORT 8080
/* El Puerto Abierto del nodo remoto */

#define MAXDATASIZE 100
/* El número máximo de datos en bytes */

int main(int argc, char *argv[])
{
    int fd, numbytes;
    /* Ficheros descriptores */

    char buf[MAXDATASIZE];
    /* En donde se almacenará el texto recibido */

    struct hostent *he;
    /* Estructura que recibirá información sobre el nodo remoto */

    struct sockaddr_in Server;
    /* Información sobre la dirección del servidor */

    if (argc !=2) {
        /* Esto es porque este programa sólo necesitará un argumento, (la IP) */
        printf("Uso: %s <Dirección IP>\n",argv[0]);
        exit(-1);
    }
}
```

```

if ((he=gethostbyname(argv[1]))==NULL){
    /* Llamada a gethostbyname() */
    printf("gethostbyname() error\n");
    exit(-1);
}

if ((fd=socket(AF_INET, SOCK_STREAM, 0))==-1){
    /* Llamada a socket() */
    printf("socket() error\n");
    exit(-1);
}

Server.sin_family = AF_INET;
Server.sin_port = htons(PORT);
/* htons() es necesaria nuevamente ;-o */
Server.sin_addr = *((struct in_addr *)he->h_addr);
/*he->h_addr pasa la información de ``*he'' a "h_addr" */
bzero(&(erver.sin_zero),8);

if(connect(fd, (struct sockaddr *)&Server,
    sizeof(struct sockaddr))==-1){
    /* Llamada a connect() */
    printf("connect() error\n");
    exit(-1);
}

if ((numbytes=recv(fd,buf,MAXDATASIZE,0)) == -1){
    /* Llamada a recv() */
    printf("Error en recv() \n");
    exit(-1);
}

buf[numbytes]='\0';

printf("Mensaje del Servidor: %s\n",buf);
/* Muestra el mensaje del servidor =) */

close(fd); /* cerramos fd =) */
}

```

El programa se compila y pasa al gumstix.

En Windows XP:

1. Cree un nuevo proyecto en C++ Builder 5, llamado sockets.
2. En la forma incluya dos Edit y un ServerSocket llamado MyServer (figura 4.2)

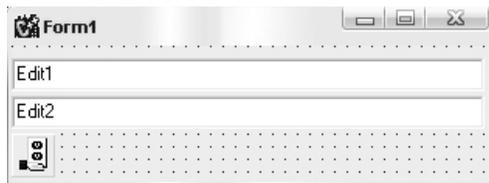


Figura 4.2 Form1 de sockets

3. En las propiedades de MyServer coloque en Port 8080.
4. En los eventos de MyServer active OnClientConnect y OnClientDisconnect
5. En los eventos de Form1 active OnCreate y OnDestroy
6. Y el código de esas funciones debe de ser el siguiente:

```
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    MyServer->Active = true;
}

void __fastcall TForm1::MyServerClientConnect(TObject *Sender,
    TCustomWinSocket *Socket)
{
    Edit2->Text= Socket->RemoteAddress + " has connect";
    Socket->SendText(Edit1->Text);
}

void __fastcall TForm1::FormDestroy(TObject *Sender)
{
    MyServer->Active = false;
}

void __fastcall TForm1::MyServerClientDisconnect(TObject *Sender,
    TCustomWinSocket *Socket)
{
    Edit2->Text= Socket->RemoteAddress + " has disconnect";
}
```

Ya están los dos programas listos, ahora se tiene que ejecutar sockets que esta en Windows. Y se ve como en la figura 4.3.



Figura 4.3 Ejecución de sockets

Ahora que ya se esta ejecutando sockets, ejecutamos en el gumstix socketc con la IP 192.168.69.2 (figura 4.4).

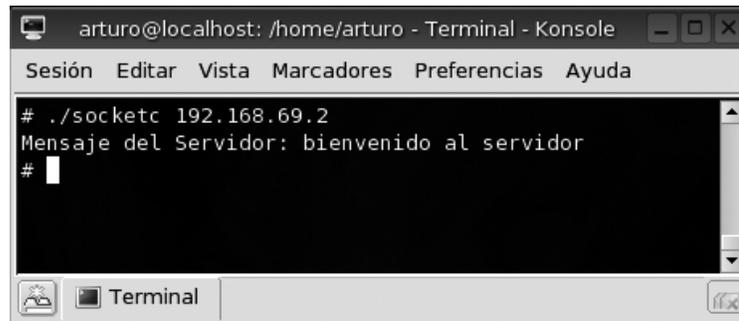


Figura 4.4 Ejecución de socketc

## 5. BIBLIOGRAFÍA

### Web

- <http://www.gumstix.org>
- <http://es.tldp.org/Tutoriales/PROG-SOCKETS/prog-sockets.html>

### Libros

- C++ Builder Developer's Guide; Jarrod Hollingworth, Dan Butterfield, Bob Swart, Jamie Allsop; Ed. SAMS 2001.

## 6. ANEXO A (Hirose 60)

Tabla 6.1 Conector Hirose 60 en Gumstix 200-bt

| Signal    | GPIO     | Pin | Signal     | GPIO     | Pin |
|-----------|----------|-----|------------|----------|-----|
| GND       |          | 1   | GND        |          | 60  |
| L_DD(9)   | GPIO(67) | 2   | USB_N      |          | 59  |
| L_BIAS    | GPIO(77) | 3   | BITCLK     | GPIO(28) | 58  |
| L_DD(3)   | GPIO(61) | 4   | L_PCLK     | GPIO(76) | 57  |
| SDATA_OUT | GPIO(30) | 5   | L_DD(15)   | GPIO(73) | 56  |
| L_DD(8)   | GPIO(66) | 6   | L_DD(14)   | GPIO(72) | 55  |
| GND       |          | 7   | FFRXD      | GPIO(34) | 54  |
| L_DD(13)  | GPIO(71) | 8   | L_DD(7)    | GPIO(65) | 53  |
| L_DD(12)  | GPIO(70) | 9   | L_DD(5)    | GPIO(63) | 52  |
| L_DD(11)  | GPIO(69) | 10  | L_DD(10)   | GPIO(68) | 51  |
| USB_P     |          | 11  | L_DD(1)    | GPIO(59) | 50  |
| IRRXD     | GPIO(46) | 12  | NACRESET   |          | 49  |
| L_DD(2)   | GPIO(60) | 13  | IR_TXD     | GPIO(47) | 48  |
| FFCTS     | GPIO(35) | 14  | L_DD(4)    | GPIO(62) | 47  |
| SDATA_IN1 | GPIO(32) | 15  | SYNC       | GPIO(31) | 46  |
| L_LCLK    | GPIO(75) | 16  | NSSP_RXD   | GPIO(84) | 45  |
| PWM(1)    | GPIO(17) | 17  | NSSP_FRAME | GPIO(82) | 44  |
| L_DD(0)   | GPIO(58) | 18  | SDATA_IN0  | GPIO(29) | 43  |
| I2C_SDA   |          | 19  | GND        |          | 42  |
| I2C_SCL   |          | 20  | L_FCLK     | GPIO(74) | 41  |
| NSSP_CLK  | GPIO(81) | 21  | L_DD(6)    | GPIO(64) | 40  |
| BT_RXD    | GPIO(42) | 22  | FFRTS      | GPIO(41) | 39  |
| NSSP_TXD  | GPIO(83) | 23  | DRQ0       | GPIO(20) | 38  |
| BT_CTS    | GPIO(44) | 24  | NRESET     |          | 37  |
| BT_TXD    | GPIO(43) | 25  | PWM(0)     | GPIO(16) | 36  |
| FFTXD     | GPIO(39) | 26  | HWTXD      | GPIO(48) | 35  |
| BT_RTS    |          | 27  | HWRTS      | GPIO(51) | 34  |
| V_BATT    |          | 28  | HWRXD      | GPIO(49) | 33  |
| V_BATT    |          | 29  | HWCTS      | GPIO(50) | 32  |
| V_BATT    |          | 30  | GND        |          | 31  |