

Integración de teléfonos móviles con GNOME vía Bluetooth

Herman Vega Jara
hvega@alumnos.ubiobio.cl
<http://gbtcr.chileforge.cl>



Temario :: Charla



- **Introducción General**

Domótica y futura integración total.
Conectividad : Cables, Irda, Bluetooth.

- **Bluetooth 802.15**

Descripción de la tecnología.
Implementaciones para Linux. BlueZ (API).

- **Software para GNOME**

SW Disponible (gnome-bluetooth).
Demo GTK+ con BlueZ.

- **Proyecto GBTcr**

Presente y Futuro.

Intro :: Domótica e Integración



- **Control Computador**

 - Sistema Reproductor Multimedia.

 - Sistema Software de presentaciones.

- **Control de Casas Inteligentes**

 - Sistema de manejo de luces.

 - Sistema de alarmas.

 - Sistema de portero.

- **Control de Autos Inteligentes**

 - Sistema Alarma.

 - Sistema GPS.

 - Sistema Check Engine.

 - Sistema DVD Player.

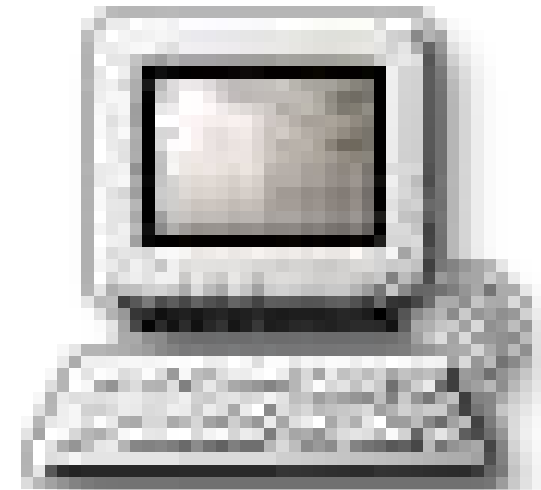
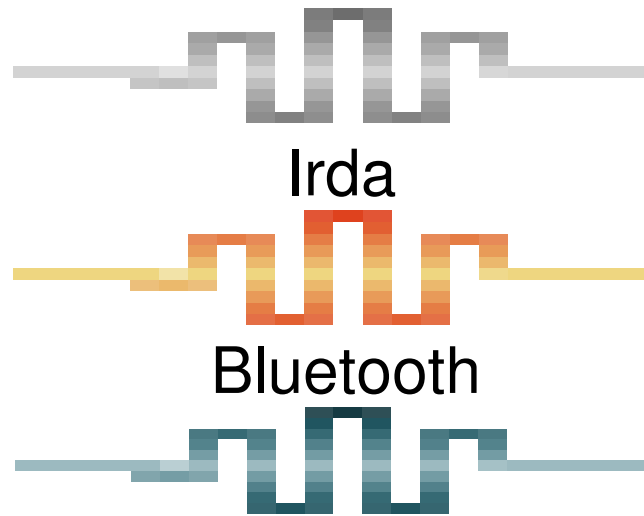
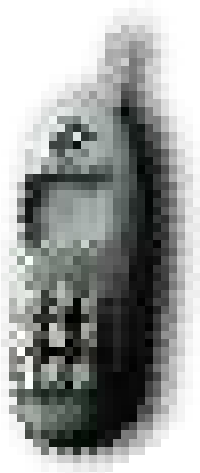
- **Integración Total**

 - Vía tecnología movil ---> Tecnología Inalámbrica.

Intro :: Escenario General de Conectividad



Cable USB, Serial



Bluetooth :: Intro



- **Descripción de la tecnología**

Estándar de protocolo universal de comunicaciones inalámbrica entre dispositivos de hardware.

Creado inicialmente por Ericsson y después estandarizado por el SIG BT (Nokia, HP, compaq, etc...) 1998

Comunicación de corto alcance (WPAN)(10mt BT v1.1 y v1.2)

Disponible mundialmente en un banda libre : ISM (2.45ghz)

Hardware de radio de bajo costo.

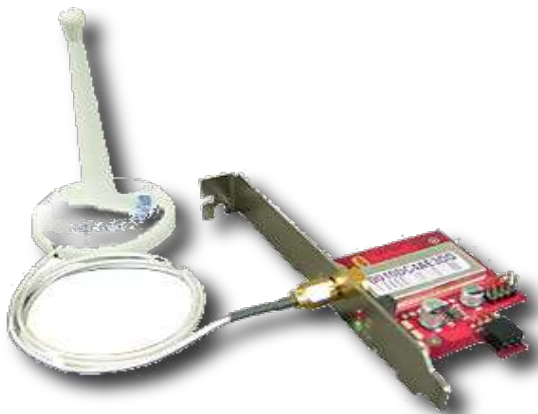
Consumo mínimo de energía.

Transmisión de voz y datos.

79 canales, 1600 hops por segundo.

(+predominancia, +seguridad, -interferencia)

Bluetooth :: Hardware



Bluetooth :: GNU/Linux



BlueZ

- libbluetooth, libbluetooth1-dev
- libsdp2, libsdp2-dev
- bluez-utils
- kernel-patch-2.6-bluez

GNOME-Bluetooth (libbtctl)

deb <http://debian.usefulinc.com/gnome> ./

deb-src <http://debian.usefulinc.com/gnome> ./

- libbtctl0, libbtctl0-dev
- python2.3-libbtctl
- gnome-bluetooth

Bluetooth :: BlueZ



Official Bluetooth Protocol Stack

Abstracción real del hardware.

Más de 150 adaptadores BT soportados.

Manejo de más de 16 adaptadores en la misma máquina.

Independiente de la arquitectura. (little endian o big endian)

Sistemas de 32 y 64 bits.

Seguridad en SMP.

Hyperthreading.

Interfaz de sockets BSD.

Diseño Modular. (kernel space, user space)

Ofrece API de programación.

Soporte BT v1.0 y 1.1 -> Planes con v1.2.

OpenSource -> Gran comunidad.



GNOME Bluetooth :: libbtctl



Subsistema Bluetooth para GNOME Orientado a Objetos.

- **GnomebtController**: Clase que controla dispositivos Bluetooth.
- Explorador de dispositivos Bluetooth (**gnome-bluetooth-manager**).
- Servidor OBEX (enviar/recibir objetos) (**gnome-obex-server**).
- Componente usado por Nautilus (**gnome-obex-send**).
 “Send via Bluetooth...”
- Proyecto ligado a **gnome-phone-manager**.

Base: biblioteca orientada a objetos **libbtctl**. Provee un acceso transparente al desarrollador a las funciones Bluetooth.

Hecha en C y con Bindings para Python (pygtk) y Mono (GTK#).

Bluetooth :: Demo



- Demo uso utilitarios de BlueZ (**users**).
 - hciconfig
 - l2ping
 - hcitool
 - hcidump
- Demo de aplicaciones existentes. (**users & developers**)
 - gnome-bluetooth
 - multisync
 - obexftp
- Uso de API de programación via BlueZ (**developers**).

BlueZ :: Demo hciconfig



```
# hciconfig -a
```

```
hci0: Type: USB
```

```
BD Address: 00:0B:0D:31:77:09 ACL MTU: 120:20 SCO MTU: 64:0
```

```
UP RUNNING PSCAN ISCAN
```

```
RX bytes:56680 acl:2276 sco:0 events:1011 errors:0
```

```
TX bytes:18552 acl:726 sco:0 commands:163 errors:0
```

```
Features: 0xff 0xff 0x05 0x38 0x18 0x18 0x00 0x00
```

```
Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
```

```
Link policy: RSWITCH HOLD SNIFF PARK
```

```
Link mode: SLAVE ACCEPT
```

```
Name: 'roncola-0'
```

```
Class: 0x3e0100
```

```
Service Classes: Networking, Rendering, Capturing
```

```
Device Class: Computer, Uncategorized
```

```
HCI Ver: 1.2 (0x2) HCI Rev: 0x0 LMP Ver: 1.2 (0x2) LMP Subver: 0x757
```

```
Manufacturer: Silicon Wave (11)
```

BlueZ :: Demo l2ping



```
# l2ping 00:0F:DE:02:88:DF
```

```
Ping: 00:0F:DE:02:88:DF from 00:0B:0D:31:77:09 (data size 20) ...
```

```
0 bytes from 00:0F:DE:02:88:DF id 0 time 20.83ms
```

```
0 bytes from 00:0F:DE:02:88:DF id 1 time 9.26ms
```

```
0 bytes from 00:0F:DE:02:88:DF id 2 time 11.25ms
```

```
0 bytes from 00:0F:DE:02:88:DF id 3 time 9.23ms
```

```
0 bytes from 00:0F:DE:02:88:DF id 4 time 11.22ms
```

```
0 bytes from 00:0F:DE:02:88:DF id 5 time 9.20ms
```

```
0 bytes from 00:0F:DE:02:88:DF id 6 time 11.18ms
```

```
7 sent, 7 received, 0% loss
```

BlueZ :: Demo hcitool



```
# hcitool info 00:0F:DE:02:88:DF
```

```
Requesting information ...
```

```
BD Address: 00:0F:DE:02:88:DF
```

```
Device Name: hvega_celu
```

```
LMP Version: 1.1 (0x1) LMP Subversion: 0x503
```

```
Manufacturer: Ericsson Technology Licensing (0)
```

```
Features: 0x04 0xca 0x31 0x00 0x00 0x00 0x00 0x00
```

```
<encryption> <RSSI> <SCO link> <u-law log> <A-law log> <CVSD>
```

```
# hcitool scan
```

```
Scanning ...
```

```
00:0F:DE:02:88:DF    hvega_celu
```

```
# hcitool cc 00:0F:DE:02:88:DF
```

```
# hcitool con
```

```
Connections:
```

```
< ACL 00:0F:DE:02:88:DF handle 5 state 1 lm MASTER
```

Libbtctl :: Demo Mono Bindings



```
using System;
using System.Runtime.InteropServices;
using Btctl;
namespace Btctl.Test {
    public class TestApp {
        [DllImport("libgobject-2.0.so")]
        static extern void g_type_init ();
        public static void Main (string[] args)
        {
            g_type_init ();
            Controller ctl = new Controller ();
            Console.WriteLine ("Inicializando Dispositivo Bluetooth...");
            if (!ctl.IsInitialised) {
                Console.WriteLine ("Error Dispositivo Bluetooth.");
                System.Environment.Exit(-1);
            }
            ctl.DeviceName += new DeviceNameHandler (Name_Device);
            Console.WriteLine ("Buscando Dispositivos Remotos...");
            ctl.DiscoverDevices ();
        }
        static void Name_Device (object obj, DeviceNameArgs args)
        {
            Console.WriteLine("Direccion de HW: {0} \t Nombre: {1}",args.Addr, args.Name);
        }
    }
}
```

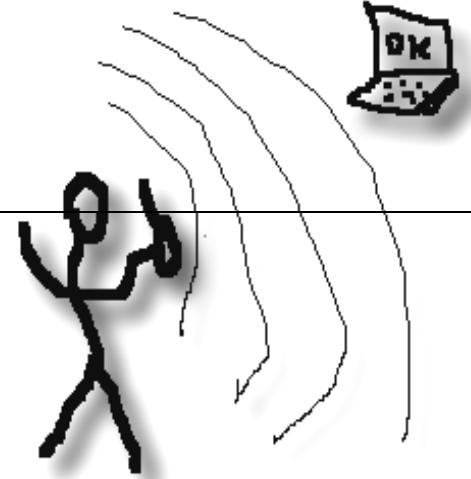
```
mcs demo_libbtctl.cs -out:demo.exe -r:btctl.dll -pkg:gtk-sharp
```

GBTcr :: Intro



- GBTcr

Proyecto de Software Libre.
GNOME Blue**T**ooth **c**ontrol **r**emoto.



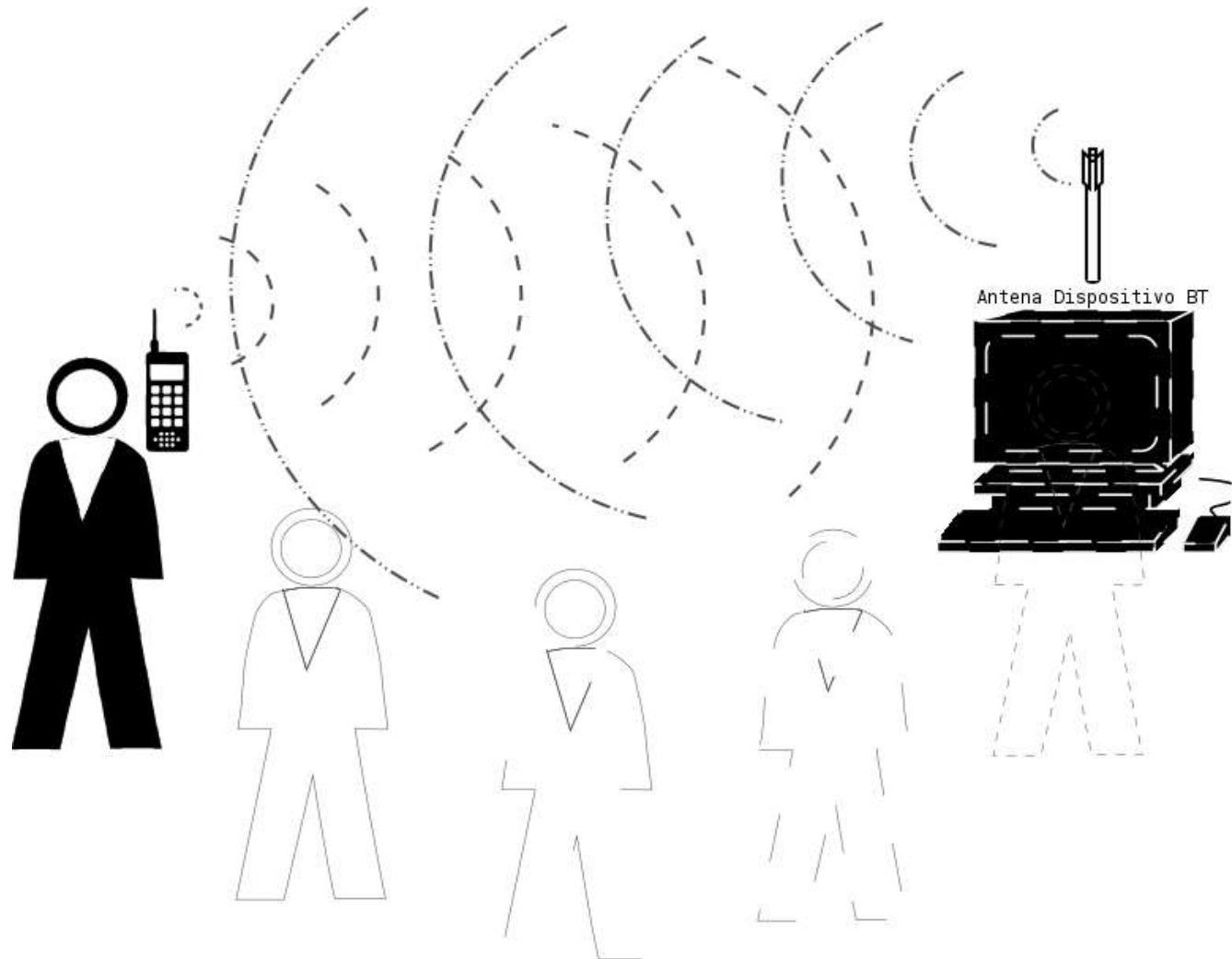
“Software de control y ejecución de ordenes de manera remota para el escritorio GNOME vía conexión inalámbrica Bluetooth”

Implementado usando: BlueZ, libbtctl, GTK+.

Origen Real: Necesidad de comodidad para el autor.

- > Tema proyecto de título.
- > RFHG 4 (noviembre 2004 en pañales)
- > RFHG 6 (junio 2005 caminando)

GBTcr :: Usuario Final

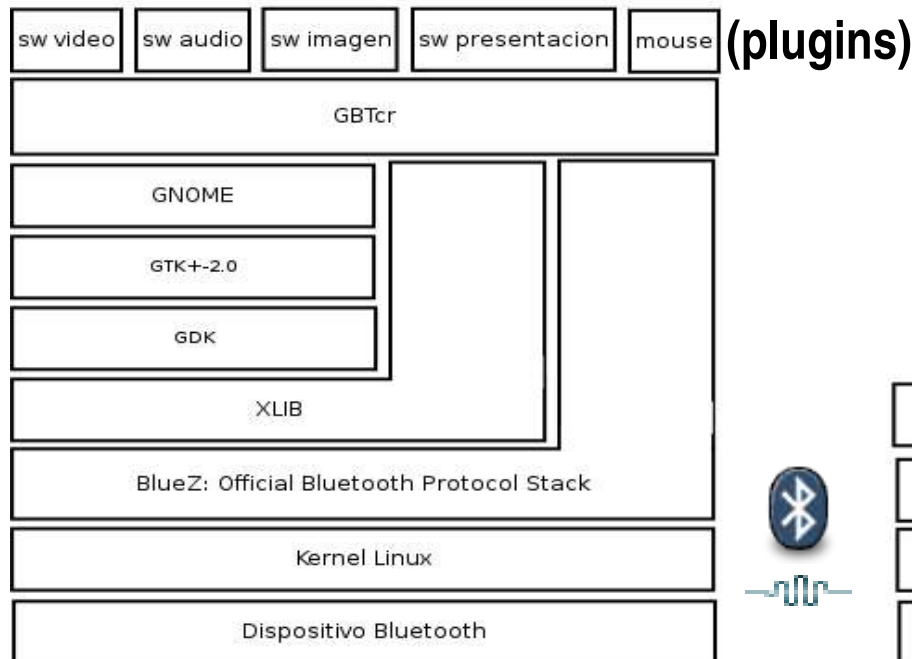


GBTcr :: Stack



Capas de software y hardware en el servidor (computador)

Capas de software y hardware en el cliente (telefono celular)



GBTcr :: Demo



GBTcr :: Demo



GBTcr :: Demo



Links :: Web<



Proyecto GBTCr

[*http://gbtcr.chileforge.cl*](http://gbtcr.chileforge.cl)

Sony Ericsson

[*http://www.sonyericsson.com/developers*](http://www.sonyericsson.com/developers)

BlueZ

[*http://www.bluez.org*](http://www.bluez.org)

GNOME-Bluetooth

[*http://www.usefulinc.com/software/gnome-bluetooth*](http://www.usefulinc.com/software/gnome-bluetooth)

Tux Movil

[*http://www.tuxmovil.org*](http://www.tuxmovil.org)

GNOME

[*http://developers.gnome.org*](http://developers.gnome.org)

Fin :: ?

¿Preguntas?

