

## Gumstix embedded plataforms

The Gumstix [1] are a very small cards [2] that can run GNU/Linux embedded systems, there are some diferent models of cards and each one has it's own conecction and expansion possibilities that supply conecction as USB-net, Bluetooth, Ethernet and/or WiFi, others supply audio I/O.



[1]: <http://www.gumstix.com>

[2]: <http://www.gumstix.com/platforms.html>

- **Basic setup**
- **Log into the Gumstix**

The first way to log into the Gumstix cards is using a cable which is a RS-232 null modem comm plugged to one of the available connectors.



The usual program to connect to the Gumstix is using kermit, it has to be configured with this instructions to get it work:

```
set line /dev/ttyS0
set speed 115200
set carrier-watch off
set handshake none
set flow-control none
set reliable
fast
set prefixing all
set file type bin
set rec pack 4096
set send pack 4096
```

This configuration can be saved into a file called `.kermrc` on your home directory which will load this configuration every time kermit is started.

The null modem configuration can be seen using the command ***show comm***.

```
C-Kermit>show comm

Communications Parameters:
Line: /dev/ttyS0, speed: 115200, mode: local, modem: generic
Parity: none, stop-bits: (default) (8N1)
Duplex: full, flow: none, handshake: none
Carrier-watch: off, close-on-disconnect: off
Lockfile: /var/lock/LCK..ttyS0
Terminal bytesize: 8, escape character: 28 (^)

Carrier Detect      (CD):  Off
Dataset Ready      (DSR):  Off
Clear To Send      (CTS):  Off
Ring Indicator     (RI):  Off
Data Terminal Ready (DTR):  0n
Request To Send    (RTS):  0n

Type SHOW DIAL to see DIAL-related items.
Type SHOW MODEM to see modem-related items.
```

With the sentence ***connect*** the connection between the Gumstix and the computer will be established.

```
C-Kermit>connect
Connecting to /dev/ttyS0, speed 115200
Escape character: Ctrl-\ (ASCII 28, FS): enabled
Type the escape character followed by C to get back,
or followed by ? to see other options.
-----

Welcome to the Gumstix Linux Distribution!
gumstix login:
```

Then we can do login using the standard *login* (root) and *password* (gumstix).

**NOTE:** To log in successfully the bluetooth module must be plugged into de basix card.

- **Cross-development tools**

[Write something]

- **Setting up interfaces:**

- **USB-net**

For this conecction we will use a USB-miniUSB cable that usually comes with the Gumstix.

It is very simply to set up a USB-net conecction using the command ***ifconfig***, it will treat this conecction as a normal net interface so the only thing to do is configure it follows:

```
# ifconfig usb0 10.0.0.1 netmask 255.255.255.0 up
```

With this sentence the USB-net usb0 interface will be set up on the Gumstix, then another USB-net interface must be set up on the computer from which the conecction will be started.

```
[root@localhost ~]# ifconfig usb0 10.0.0.2 netmask 255.255.255.0 up
```

The ***ifconfig*** command will show on both, the Gumstix and the localhost, the configuration of all interfaces.

On the Gumstix:

```
# ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:13 errors:0 dropped:0 overruns:0 frame:0
        TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:4329 (4.2 KiB)  TX bytes:4329 (4.2 KiB)

usb0    Link encap:Ethernet  HWaddr 0A:00:FD:42:F3:21
        inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:6 errors:0 dropped:0 overruns:0 frame:0
        TX packets:216 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:348 (348.0 B)  TX bytes:127440 (124.4 KiB)
```

On the localhost:

```
[root@localhost ~]# ifconfig
eth0    Link encap:Ethernet  HWaddr 00:50:DA:43:DA:22
        inet addr:194.182.170.122  Bcast:194.182.170.255  Mask:255.255.255.0
        inet6 addr: fe80::250:daff:fe43:da22/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:733584 errors:0 dropped:0 overruns:0 frame:0
        TX packets:470003 errors:0 dropped:0 overruns:0 carrier:0
        collisions:7638 txqueuelen:1000
        RX bytes:639207746 (609.5 MiB)  TX bytes:135037402 (128.7 MiB)
        Interrupt:11 Base address:0xdc80

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:2379 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2379 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:3808422 (3.6 MiB)  TX bytes:3808422 (3.6 MiB)

usb0    Link encap:Ethernet  HWaddr 46:59:DE:1D:71:EC
        inet addr:10.0.0.2  Bcast:10.0.0.255  Mask:255.255.255.0
        inet6 addr: fe80::4459:deff:fe1d:71ec/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:380 errors:0 dropped:0 overruns:0 frame:0
        TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:218880 (213.7 KiB)  TX bytes:432 (432.0 b)
```

**NOTE:** The localhost Linux will not recognise usb0 as an interface until the Gumstix has booted.

```
[root@localhost ~]# ifconfig usb0 10.0.0.2 netmask 255.255.255.0 up
SIOCSIFADDR: No such device
usb0: unknown interface: No such device
SIOCSIFNETMASK: No such device
SIOCGIFADDR: No such device
SIOCSIFBROADCAST: No such device
usb0: unknown interface: No such device
```

To test the configuration you can ping on both sides:

```
# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=64 time=8.2 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=5.0 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=4.6 ms

--- 10.0.0.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 4.6/5.9/8.2 ms
```

```
[root@localhost ~]# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=0 ttl=64 time=2.03 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.32 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.21 ms

--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.217/1.524/2.036/0.365 ms, pipe 2
```

With this simple two commands a USB-net connection will be created between the computer and the card.

- **Ethernet**

For this connection we will use a RJ-45 straight through cable.

Setting up an ethernet connection is as simple as the USB-net one.

Using **ifconfig** the connection can be established as follows:

```
# ifconfig eth0 192.168.0.201 netmask 255.255.255.0 up
```

```
[root@localhost ~]# ifconfig eth1 192.168.0.200 netmask 255.255.255.0 up
```

If you have more than one ethernet card you will have to decide where do you want to establish the connection so maybe you will have to change eth0 to eth1, eth2, etc.

Using **ifconfig** with no arguments you can see which interfaces are up on your machine.

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:DA:43:DA:22
          inet addr:194.182.170.122  Bcast:194.182.170.255  Mask:255.255.255.0
          inet6 addr: fe80::250:daff:fe43:da22/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:743624 errors:0 dropped:0 overruns:0 frame:0
          TX packets:476039 errors:0 dropped:0 overruns:0 carrier:0
          collisions:7665 txqueuelen:1000
          RX bytes:642375402 (612.6 MiB)  TX bytes:135926324 (129.6 MiB)
          Interrupt:11 Base address:0xdc80

eth1      Link encap:Ethernet  HWaddr 00:06:5B:64:4B:8B
          inet addr:192.168.0.200  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::206:5bff:fe64:4b8b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:152 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          RX bytes:31226 (30.4 KiB)  TX bytes:5699 (5.5 KiB)
          Interrupt:11 Base address:0xdc00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2383 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2383 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3808758 (3.6 MiB)  TX bytes:3808758 (3.6 MiB)
```

**NOTE:** to establish succesful conecction between Gumstix and other machines all IPs have to be from the same LAN range, remember that the three possible ranges for LAN are:

- 192.168.x.x
- 172.16.x.x
- 10.x.x.x

Now you can **ping** both interfaces to verify that the connection has been set up.

```
# ping 192.168.0.200
PING 192.168.0.200 (192.168.0.200): 56 data bytes
64 bytes from 192.168.0.200: icmp_seq=0 ttl=64 time=2.3 ms
64 bytes from 192.168.0.200: icmp_seq=1 ttl=64 time=0.5 ms
64 bytes from 192.168.0.200: icmp_seq=2 ttl=64 time=0.5 ms

--- 192.168.0.200 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.5/1.1/2.3 ms
```

```
[root@localhost ~]# ping 192.168.0.201
PING 192.168.0.201 (192.168.0.201) 56(84) bytes of data.
64 bytes from 192.168.0.201: icmp_seq=0 ttl=64 time=1.44 ms
64 bytes from 192.168.0.201: icmp_seq=1 ttl=64 time=0.481 ms
64 bytes from 192.168.0.201: icmp_seq=2 ttl=64 time=0.472 ms

--- 192.168.0.201 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.472/0.799/1.445/0.457 ms, pipe 2
```

- **Bluetooth**

Bluetooth connections have to be handled in a different way as the net connections.

To set up a bluetooth connection you will have to use this commands/services:

<i>hciconfig</i>	(Host Controller Interface) for configure bluetooth devices
<i>hcidtool</i>	for configure bluetooth devices and send some special commands
<i>sdptool</i>	for control and interrogate SDP servers
<i>sdptool</i>	L2CAP (Logical Link Control and Adaptation Protocol, bluetooth protocol stack) echo request and receive answer
<i>sdpd</i>	SDP Daemon
<i>pand</i>	Bluetooth PAN (Personal Area Network) Daemon

The kind of bluetooth protocol connection is master/slave, which means that one of the devices must be configured as master, with it's special features, and the other must as the slave, with it's own special features, so the connection will be point-to-point and it will only involve two devices, unless one of them will act as a proxy/gateway forwarding packages over a net connection.

Then we have two simple connection possibilities:

1. Gumstix - Gumstix
2. Gumstix - Localhost

And one multiple connection:

3. Gumstix - Localhost - Gumstix

- **Gumstix - Gumstix bluetooth connection**

First you have to check that your system recognizes the bluetooth device, **hciconfig** will show the bluetooth devices attached to the Gumstix. If **hciconfig** don't return any device, it fails or the returned bluetooth address is 00:00:00:00:00:00 it means that there is something wrong, could be the bluetooth card or the buildroot version.

Bluetooth device on Gumstix:

```
# hciconfig
hci0:  Type: UART
      BD Address: 00:80:37:27:1C:8A ACL MTU: 339:7 SCO MTU: 120:6
      DOWN
      RX bytes:7461 acl:0 sco:0 events:1062 errors:0
      TX bytes:9749 acl:0 sco:0 commands:1062 errors:0
```

Bluetooth device on Gumstix (MAC error):

```
hci0:  Type: UART
      BD Address: 00:00:00:00:00:00 ACL MTU: 0:0 SCO MTU: 0:0
      DOWN
      RX bytes:0 acl:0 sco:0 events:0 errors:0
      TX bytes:4 acl:0 sco:0 commands:1 errors:0
```

If everthings goes right we will have on both Gumstix a bluetooth device, we must set up it:

```
# hciconfig hci0 up
# hciconfig
hci0:  Type: UART
      BD Address: 00:80:37:27:1C:8A ACL MTU: 339:7 SCO MTU: 120:6
      UP RUNNING PSCAN ISCAN INQUIRY
      RX bytes:7589 acl:0 sco:0 events:1077 errors:0
      TX bytes:10076 acl:0 sco:0 commands:1077 errors:0
```

Then we can scan all bluetooth devices in our range with the tool **hcitool**:

```
# hcitool scan
Scanning ...
    00:0F:DE:25:D1:B5      K700i
    00:80:37:20:D2:51      Gumstix (0)
    00:10:C6:94:FA:24      D610-009
```

The **l2ping** command is like a **ping** but for bluetooth devices, it works with bluetooth **MAC** address instead of ip addresses as the usual **ping** does.

We can try **l2ping** both devices from each other.

```
# l2ping 00:80:37:20:D2:51
Ping: 00:80:37:20:D2:51 from 00:80:37:27:1C:8A (data size 44) ...
44 bytes from 00:80:37:20:D2:51 id 0 time 186.91ms
44 bytes from 00:80:37:20:D2:51 id 1 time 92.92ms
44 bytes from 00:80:37:20:D2:51 id 2 time 76.73ms
44 bytes from 00:80:37:20:D2:51 id 3 time 96.81ms
4 sent, 4 received, 0% loss
```

```
# l2ping 00:80:37:27:1C:8A
Ping: 00:80:37:27:1C:8A from 00:80:37:20:D2:51 (data size 20) ...
20 bytes from 00:80:37:27:1C:8A id 0 time 230.25ms
20 bytes from 00:80:37:27:1C:8A id 1 time 116.97ms
20 bytes from 00:80:37:27:1C:8A id 2 time 152.04ms
20 bytes from 00:80:37:27:1C:8A id 3 time 82.09ms
4 sent, 4 received, 0% loss
```

More information about the bluetooth devices can be obtained using the command **sdptool browse**



```
# sdptool browse
Inquiring ...
Browsing 00:80:37:27:1C:8A ...
Service Name: Serial Port
Service Description: COM Port
Service RecHandle: 0x10000
Service Class ID List:
  "Serial Port" (0x1101)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1
Language Base Attr List:
  code_IS0639: 0x656e
  encoding: 0x6a
  base_offset: 0x100
Profile Descriptor List:
  "Serial Port" (0x1101)
    Version: 0x0100
```

Now start the *SDP daemon*:

```
# sdpd
```

On the Gumstix that is going to be the Master side:

```
# pand --listen --role NAP --master --autozap
```

The Gumstix slave side has to be configured to listen for connections using **pand** with the option **-s** or **--listen**.

```
# pand -s
```

On the Gumstix that is going to be the Master side you'll have to set the hardware address of the slave Gumstix using **pand**:

```
# pand --connect 00:80:37:20:D2:51 --service NAP --autozap
```

At this time you must have a bnep (bluetooth network) set up, to check it use **pand -l** on both sides:

```
# pand -l
bnep0 00:80:37:20:D2:51 NAP
```

```
# pand -l
bnep0 00:80:37:27:1C:8A PANU
```

To set up the TCP/IP over Bluetooth you just need to configure both devices with **ifconfig** as always:

```
# ifconfig bnep0 10.1.5.1 netmask 255.255.255.0 up
```

```
# ifconfig bnep0 10.1.5.2 netmask 255.255.255.0 up
```

And check that ifconfig returns the bnep0 network on both sides:

```

# ifconfig
bnep0    Link encap:Ethernet  HWaddr 00:80:37:27:1C:8A
         inet addr:10.1.5.1  Bcast:10.255.255.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:148 (148.0 B)  TX bytes:35813 (34.9 KiB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING  MTU:16436  Metric:1
         RX packets:12 errors:0 dropped:0 overruns:0 frame:0
         TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:4117 (4.0 KiB)  TX bytes:4117 (4.0 KiB)

usb0    Link encap:Ethernet  HWaddr D6:02:33:43:C5:E1
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

```

```

# ifconfig
bnep0    Link encap:Ethernet  HWaddr 00:80:37:20:D2:51
         inet addr:10.1.5.2  Bcast:10.255.255.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:60 errors:0 dropped:0 overruns:0 frame:0
         TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:35248 (34.4 KiB)  TX bytes:128 (128.0 B)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         UP LOOPBACK RUNNING  MTU:16436  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

usb0    Link encap:Ethernet  HWaddr 06:00:FC:42:C2:61
         inet addr:10.0.0.3  Bcast:0.0.0.0  Mask:255.255.255.0
         UP BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

```

Now you can test your connection using *ping*:

```

# ping 10.1.5.2
PING 10.1.5.2 (10.1.5.2): 56 data bytes
64 bytes from 10.1.5.2: icmp_seq=0 ttl=64 time=205.9 ms
64 bytes from 10.1.5.2: icmp_seq=1 ttl=64 time=96.9 ms
64 bytes from 10.1.5.2: icmp_seq=2 ttl=64 time=109.6 ms

--- 10.1.5.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 96.9/137.4/205.9 ms

```

```
# ping 10.1.5.1
PING 10.1.5.1 (10.1.5.1): 56 data bytes
64 bytes from 10.1.5.1: icmp_seq=0 ttl=64 time=143.2 ms
64 bytes from 10.1.5.1: icmp_seq=1 ttl=64 time=197.1 ms
64 bytes from 10.1.5.1: icmp_seq=2 ttl=64 time=159.7 ms

--- 10.1.5.1 ping statistics ---
5 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 143.2/166.6/197.1 ms
```

Sometimes we get some error messages:

```
l2cap_rcv_acldata: Unexpected start frame (len 589)
l2cap_rcv_acldata: Unexpected start frame (len 589)
l2cap_rcv_acldata: Unexpected start frame (len 589)
l2cap_rcv_acldata: Unexpected start frame (len 589)
l2cap_rcv_acldata: Unexpected start frame (len 589)
```

At this moment we don't know what does it mean, sometimes the bluetooth connection is slower or the interfaces loose few more packets.

- **Gumstix – Localhost bluetooth connection**

If the bluetooth configuration on the gumstix has been tested before that step will be skipped, but to get the bluetooth working on the localhost we have to load bluetooth modules.

To configure a Belkin USB-Bluetooth under Fedora Core 4 we will follow the next steps:

1. `lsusb`
2. `rpm -q -a | grep -i bluez`
3. `/etc/init.d/bluetooth start`

We have to check whether if we have a device called 'rfcomm0' on /dev/ or not, if not we need to create it:

```
#mknod /dev/rfcomm0 c 216 0
```

And then we just follow the same steps as setting up a bluetooth connection over two gumstix.

[EXAMPLES]

Referrals:

[1]: <http://blog.lobstertechnology.com/2006/02/02/>

[2]: <http://em.typodemon.com/wordpress/?p=31>

- **Transferring files to Gumstix**
- **Web server setup**
- **Reflashing**
- **Comments**

As Red Hat Fedora Core doesn't log the kernel messages maybe you would like to change the configuration to see the messages that the Gumstix passes to the localhost or viceversa, on the file:

```
/etc/syslog.conf
```

And replace the line that refers to the kernel linking to a file that you can later read:

```
#kern.* /dev/console  
kern.* /var/log/kernel
```