

# *Openmoko: il GSM libero!*

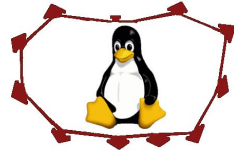


# openmoko

di

Rodolfo Giometti

<giometti@enneenne.com>



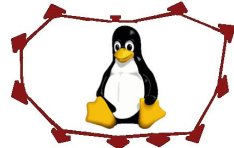
# Openmoko & Neo1973

Openmoko è la distribuzione che gira (solo per ora) sul telefono cellulare Neo1973.

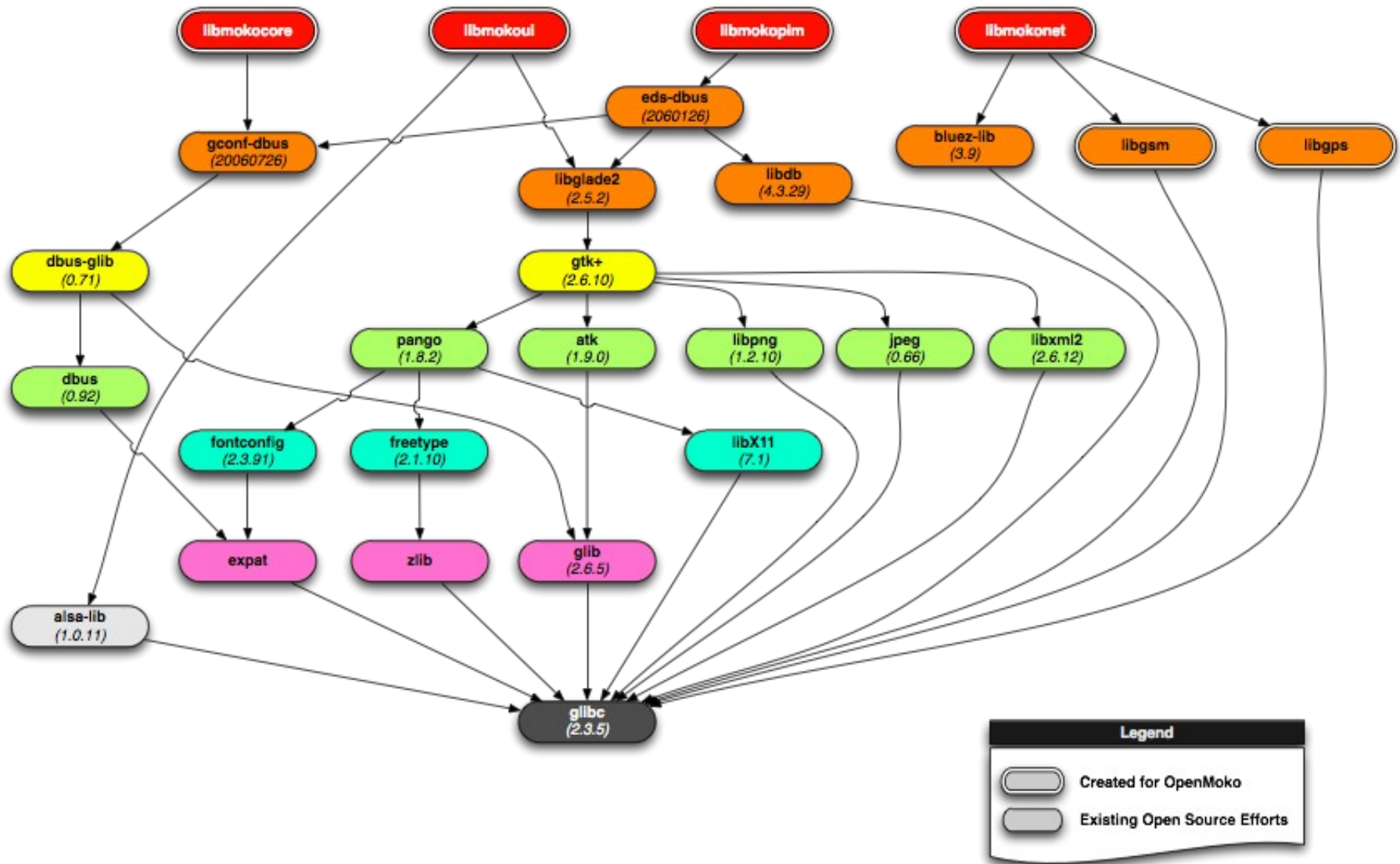


# *Le caratteristiche di Openmoko*

- Definizione di proprie librerie per la gestione del telefonino (lista contatti, storico delle chiamate, ecc.) denominate «**libmoko**».
- Uso di **GTK+2** per la parte grafica.
- Uso di **Alsa-Lib** per la parte audio.
- Uso di **dbus** per i messaggi tra applicazioni.
- Uso di **Gconf** per memorizzare le preferenze e le sessioni.
- Uso di **Bluez-lib** per la gestione delle connessioni bluetooth.
- Uso di **libgsm** per la gestione del GSM.
- Uso di **libgps** per la gestione del GPS.



# La struttura delle librerie di Openmoko

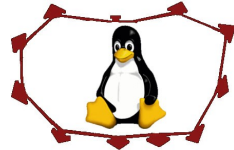


# *Le caratteristiche principali del Neo1973*

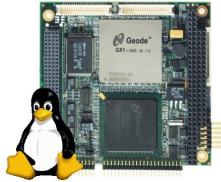
- CPU Samsung S3C2410AL-26 (fino a 266MHz)
- 128 MB RAM – 64 MB Flash
- TFT 480x640 65K colori con touch screen
- GSM – GPRS
- Bluetooth – USB (device)
- MicroSD
- Audio stereo AC97 – Antenna GPS
- Vibratore di chiamata – Amplificatore stereo



- U-Boot 1.2.x
- Linux 2.6.21
- Ambiente GNU basato su OpenEmbedded



# Prima di tutto un po' di nomi...



Con il termine *target* si intende il sistema *embedded* sul quale gira la distribuzione *embedded*, ed è quindi il sistema finale da sviluppare.



Con il termine *host* si intende il sistema di sviluppo, cioè è la macchina che utilizziamo per *cross-compilare* e che ci aiuta nello sviluppo.



La *toolchain* è invece l'insieme delle *binutils*, delle librerie (libc, ecc.) e del *cross-compilatore*.



Il *JTAG* è un dispositivo con il quale è possibile realizzare sessioni di debug molto spinte (si può mandare la CPU passo passo!)

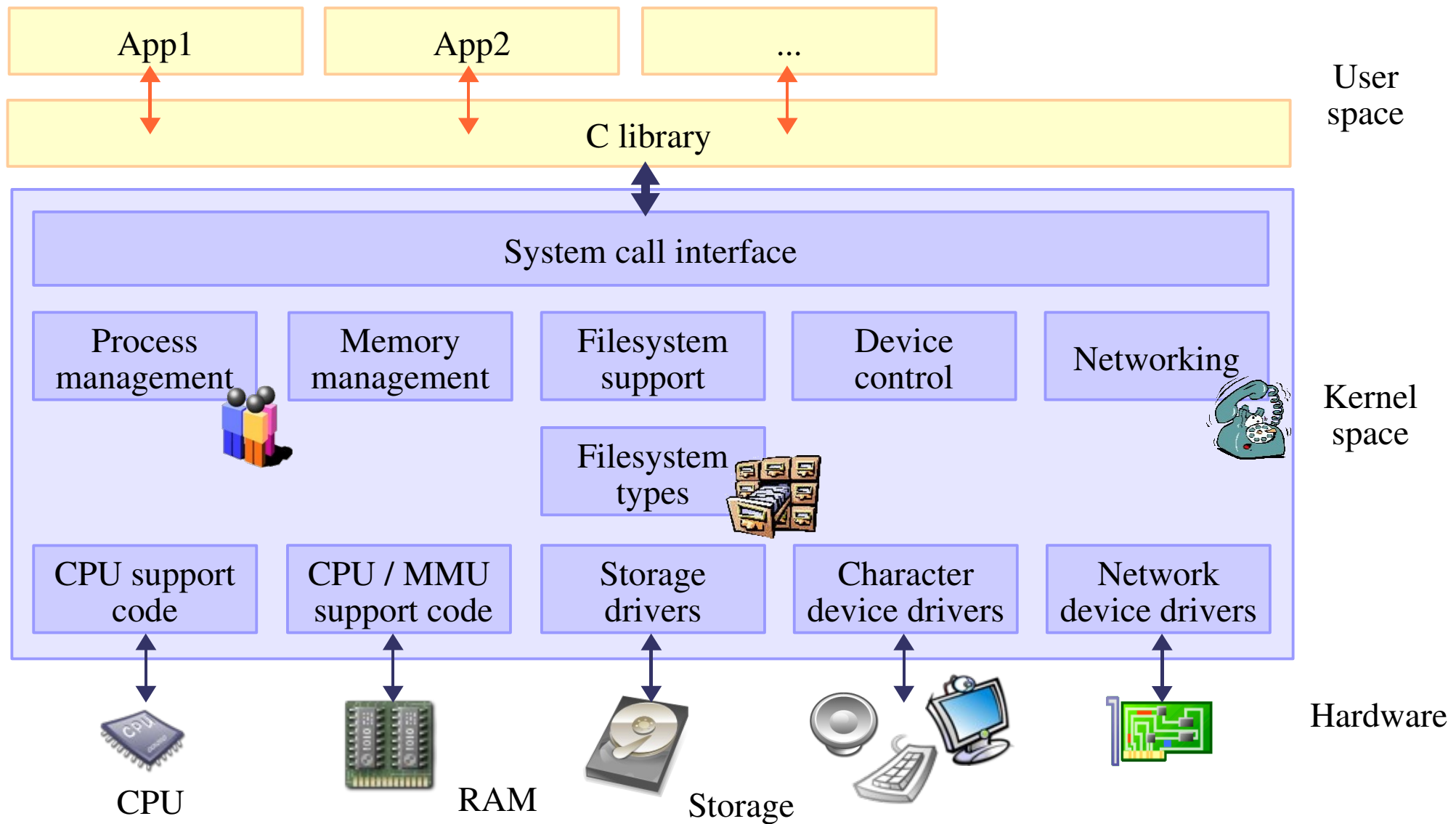
# Il sistema GNU/Linux

Parliamo di *GNU/Linux* e non solo di *Linux* perché *Linux* di per se è solo il nucleo del sistema.



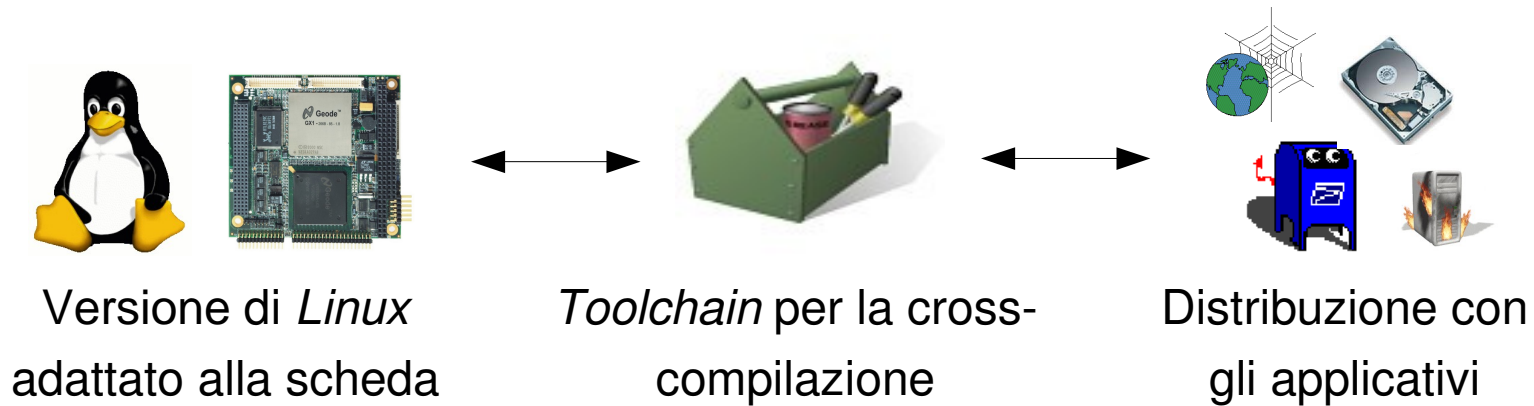
Quando parliamo di «sistema operativo» intendiamo quindi l'insieme del nucleo (*Linux*) più tutta una serie di applicazioni di gestione, utilità, ecc. (*GNU*) del sistema stesso.

# Il nucleo (o kernel)



# Sistema di sviluppo embedded

Quando scarichiamo o acquistiamo una distribuzione *embedded GNU/Linux* solitamente ci vengono fornite diverse componenti.



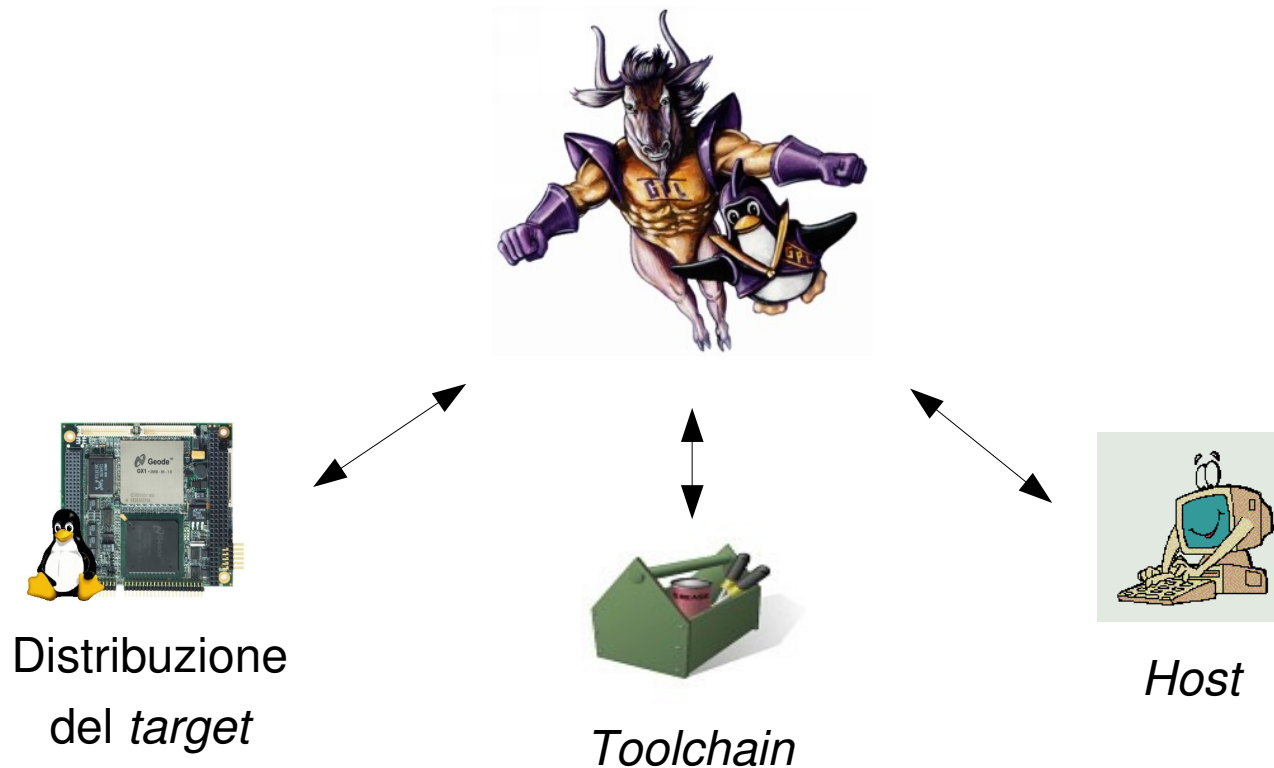
Se questo non avviene allora occorre *arrangiarsi*:



- Prendere i sorgenti
- Applicare le *patch*
- Trovare una *toolchain* & distribuzione
- Compilare, compilare, compilare...

# Distribuzioni GNU/Linux

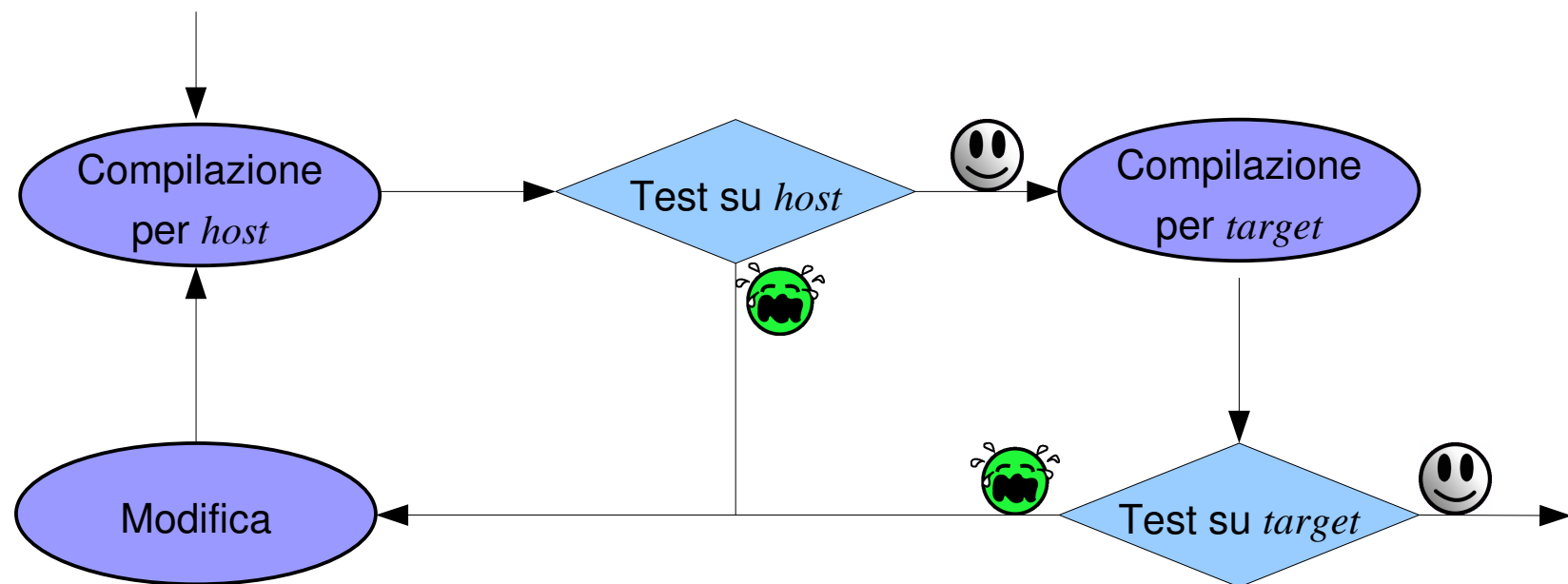
Nel caso dei sistemi *embedded* generalmente possiamo parlare di «distribuzione *embedded GNU/Linux*» poiché abbiamo non solo il sistema *target* di quel tipo ma anche i *tool di sviluppo* appartengono al progetto *GNU* (e molte volte anche il sistema *host*).



# Lo sviluppo passo passo

Ogni applicativo è una parte a se stante e quindi è facilmente rimpiazzabile sia in fase di sviluppo che in una successiva fase di messa in opera.

addirittura la fase di test e validazione potrebbe essere effettuata sul sistema *host* e poi, semplicemente ricompilando il tutto trasferita sul sistema *target*!



# Dove trovare aiuto!!!

## Use the source, Luke!

Thanks to LucasArts



Leggere il codice è importantissimo, ma le fonti di informazione possono essere molte a partire dalla rete.



Ci sono poi i libri specifici per la programmazione in user o kernel space.



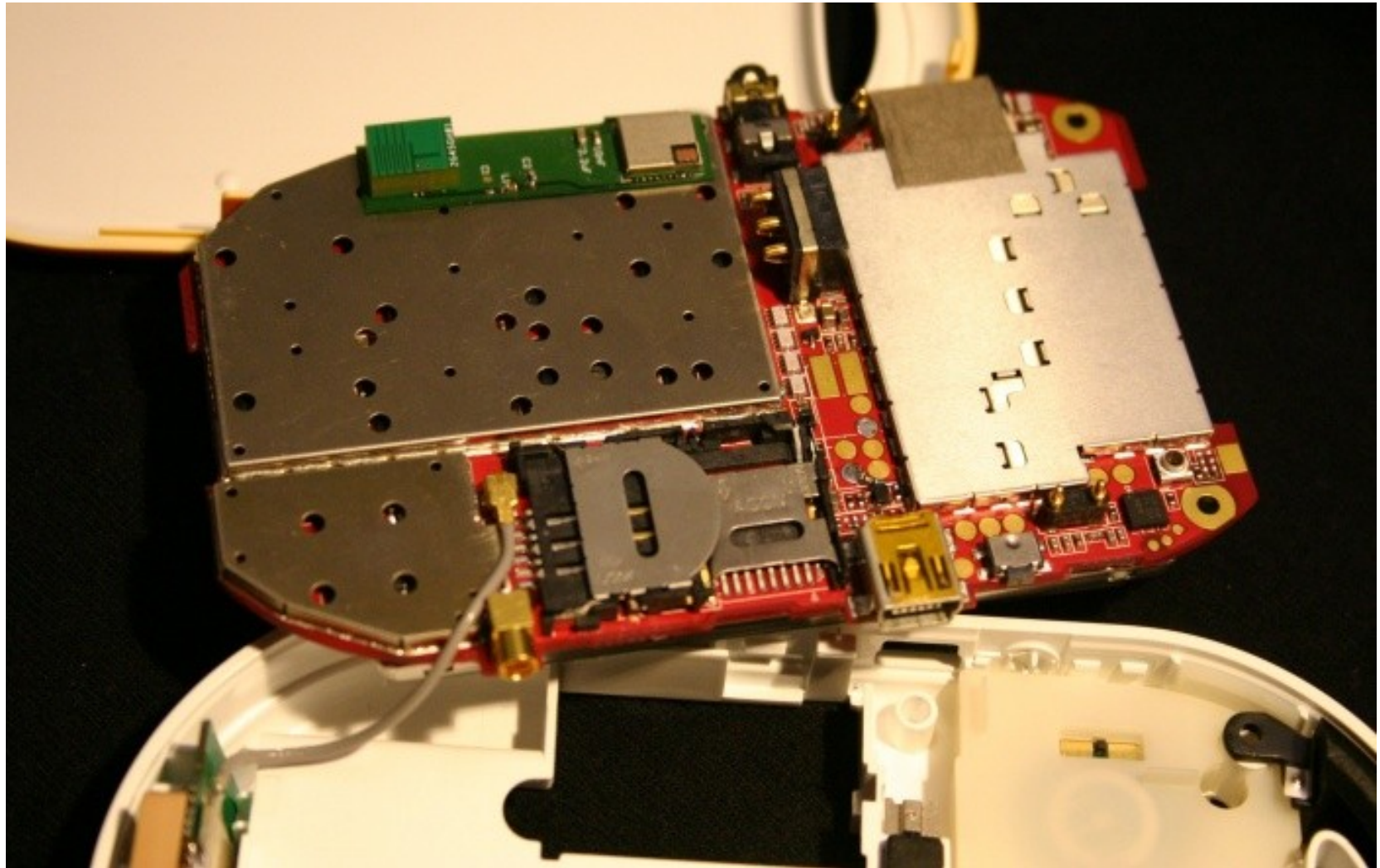
E in fine ci sono anche i consulenti...



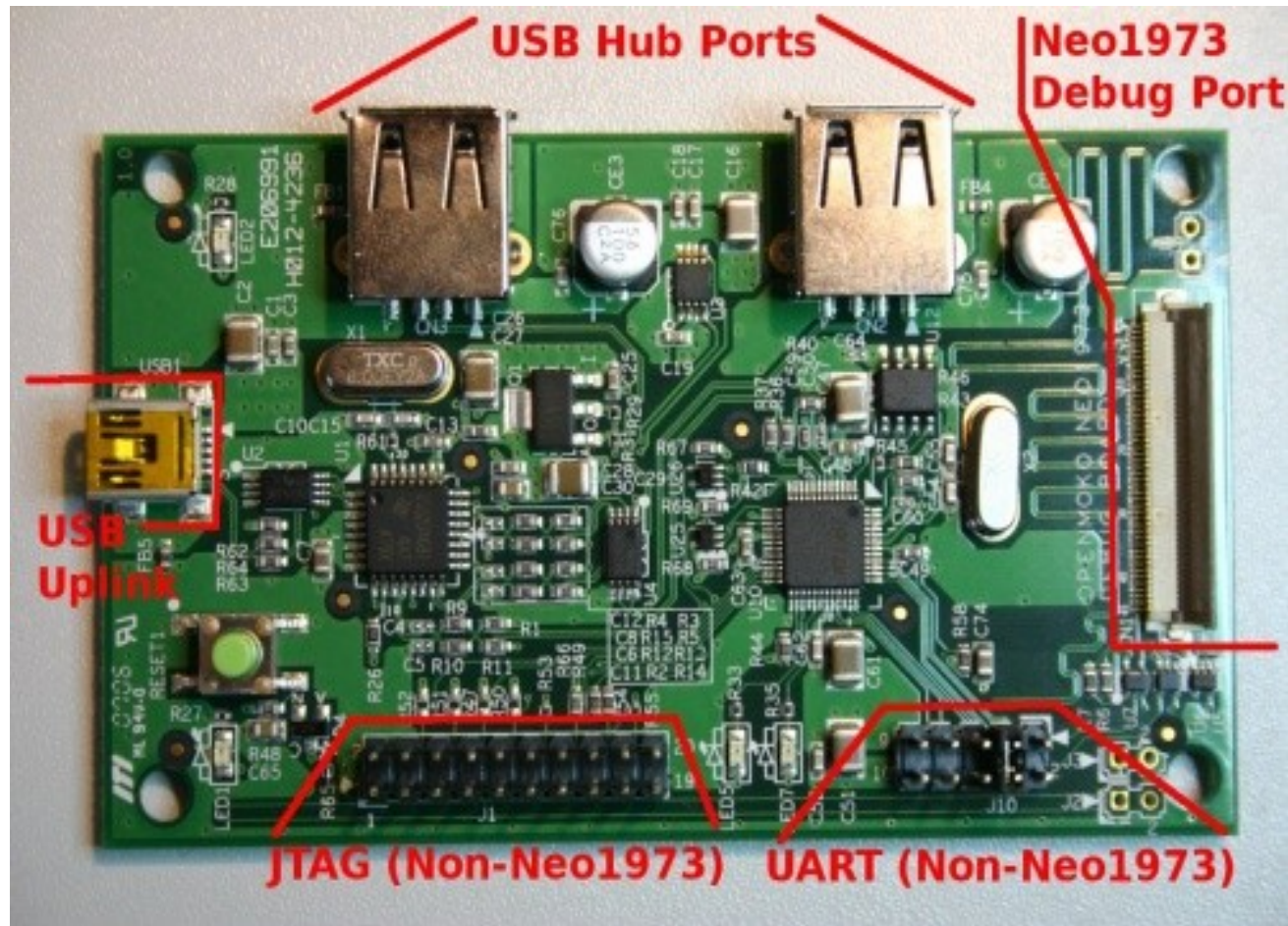
# *Il Neo1973 visto da fuori...*



*... e da dentro*

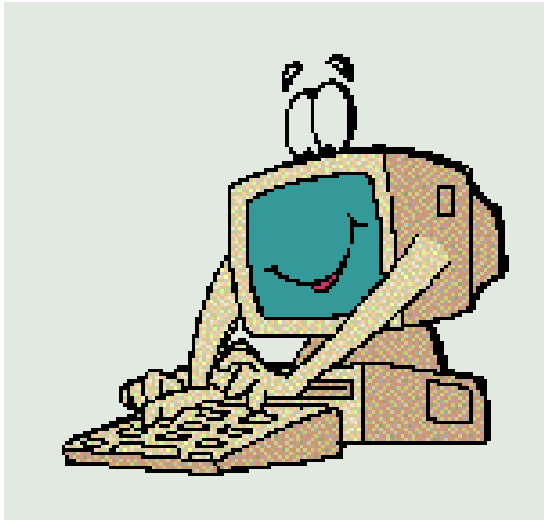


# La scheda di DEBUG



# Programmare la flash

Openmoko utilizza un interessante modo per l'aggiornamento della memoria flash: USB DFU (USB Device Firmware Update) protocol.



# *Il menu di u-boot*

Inoltre U-boot (il boot loader) è stato modificato per implementare un menu di gestione minimale del sistema.



# Collegamenti all'host

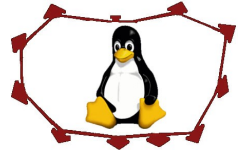
È possibile collegare il Neo1973 all'host via USB serial o ethernet.

Connettere il dispositivo e caricare il modulo `usbnet`, quindi configurare l'interfaccia `usb0`:

```
ifconfig usb0 192.168.0.200 netmask 255.255.255.0
```

Quindi configurare la propria interfaccia (es. 192.168.0.105):

1. `ping -I usb0 192.168.0.202`
2. `route add -host 192.168.0.202/32 dev usb0`
3. `ssh root@192.168.0.202`



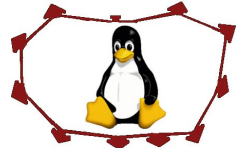
# Gestire il sistema

Essendo basata su OpenEmbedded, Openmoko supporta il sistema di aggiornamento pacchetti `ipkg`.

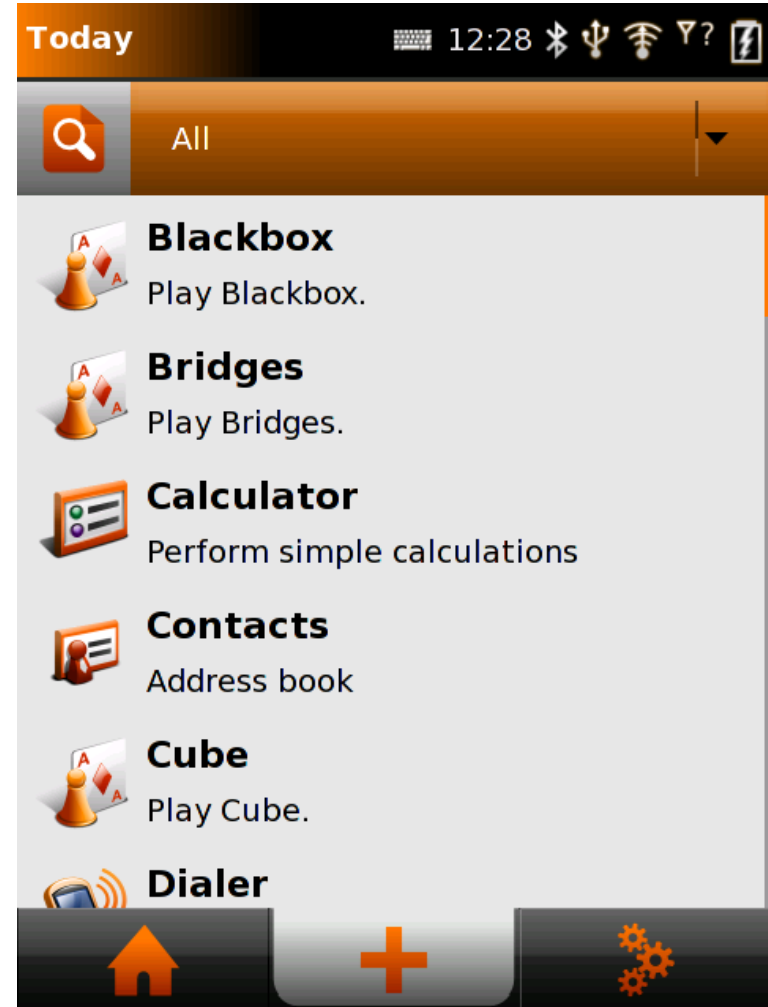
Basta aggiungere in `/etc/ipkg/base-feed.conf` gli indirizzi degli archivi e quindi usare i comandi:

```
$ ipkg update && ipkg upgrade
Downloading http://openmoko.org/ipk/Packages.gz
Inflating http://openmoko.org/ipk/Packages.gz
Updated list of available packages in /var/lib/ip
kg/snapshot-all
```

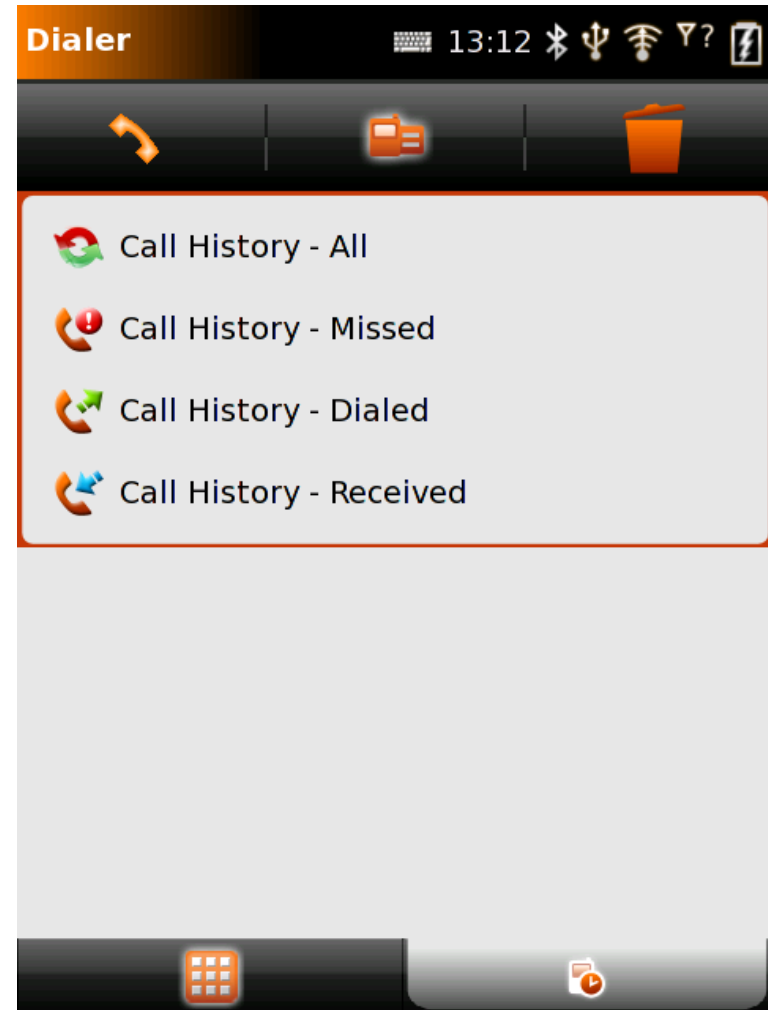
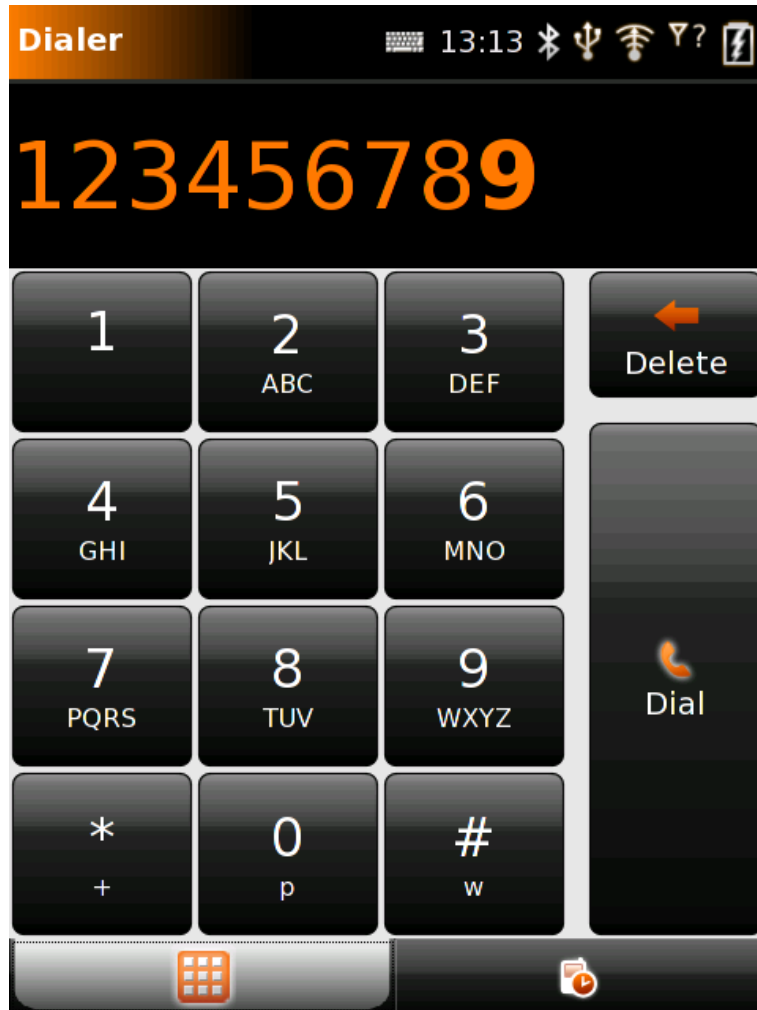
Nota: gli indirizzi degli archivi sono stati «accorciati» per problemi di spazio.



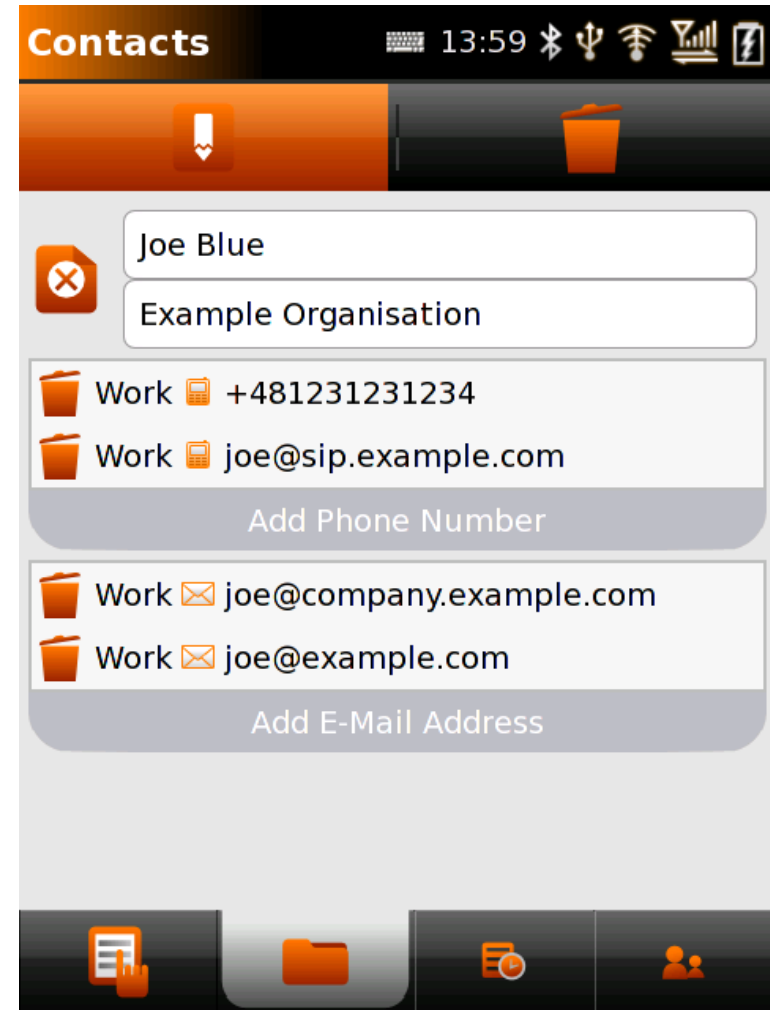
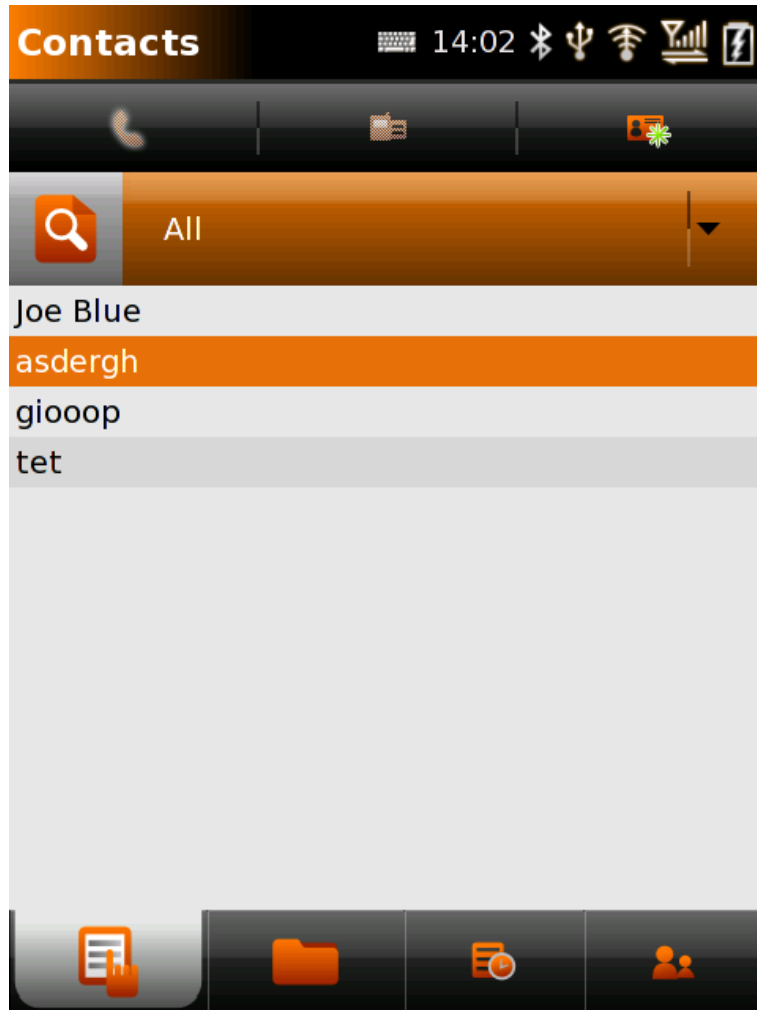
# Alcuni screenshot: Today



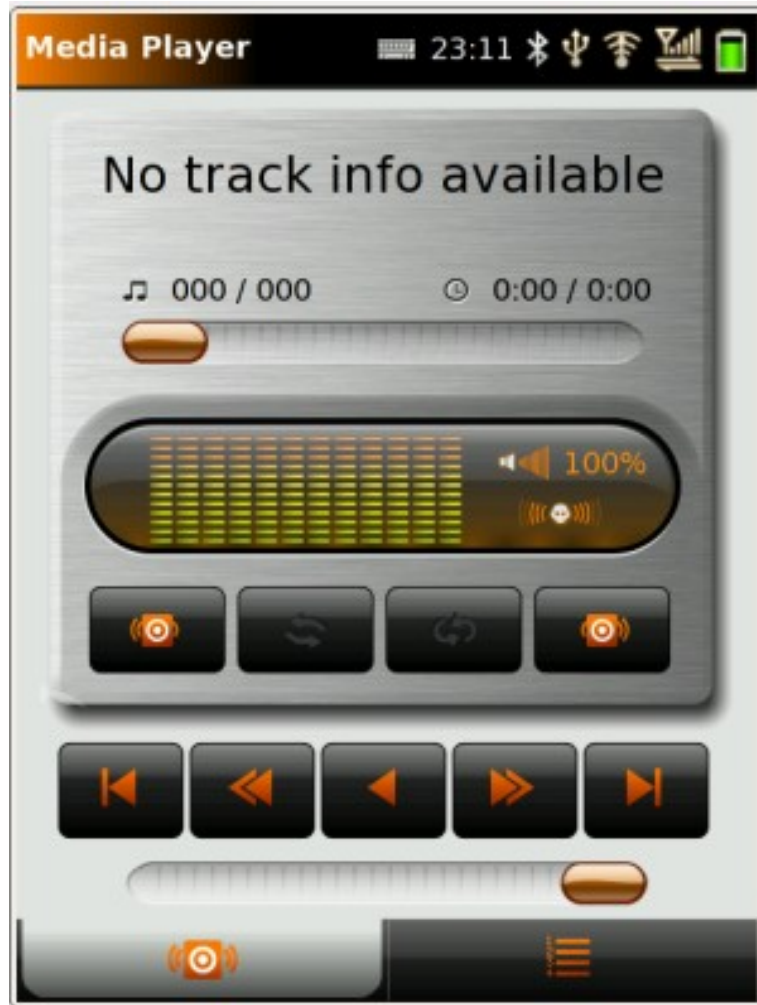
# Alcuni screenshot: Dialer



# Alcuni screenshot: Contacts



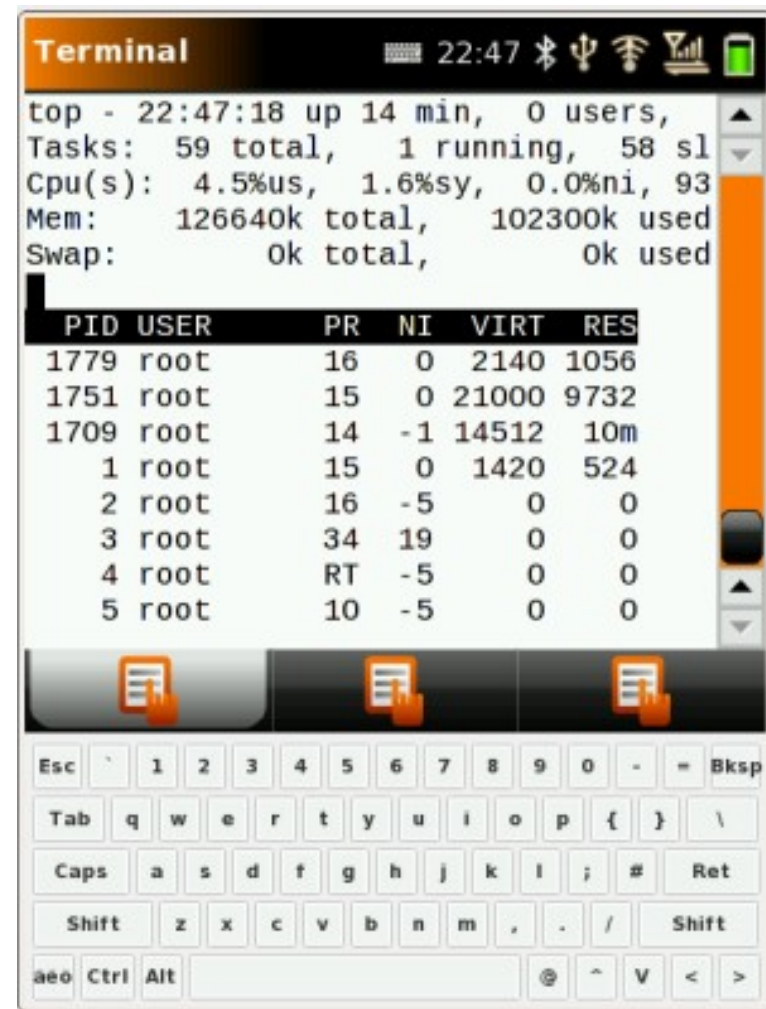
# Alcuni screenshot: Amusements



# Alcuni screenshot: System



Terminal window showing a command prompt. The terminal title is "Terminal" and the time is 22:39. The prompt is root@fic-gta01:/\$. The command being entered is `wget commandline.org.uk`. The terminal has a scroll bar on the right and a keyboard overlay at the bottom.



Terminal window showing the output of the `top` command. The terminal title is "Terminal" and the time is 22:47. The output shows system statistics and a list of running processes.

```
top - 22:47:18 up 14 min, 0 users,
Tasks: 59 total, 1 running, 58 sl
Cpu(s): 4.5%us, 1.6%sy, 0.0%ni, 93
Mem: 126640k total, 102300k used
Swap: 0k total, 0k used
```

| PID  | USER | PR | NI | VIRT  | RES  |
|------|------|----|----|-------|------|
| 1779 | root | 16 | 0  | 2140  | 1056 |
| 1751 | root | 15 | 0  | 21000 | 9732 |
| 1709 | root | 14 | -1 | 14512 | 10m  |
| 1    | root | 15 | 0  | 1420  | 524  |
| 2    | root | 16 | -5 | 0     | 0    |
| 3    | root | 34 | 19 | 0     | 0    |
| 4    | root | RT | -5 | 0     | 0    |
| 5    | root | 10 | -5 | 0     | 0    |

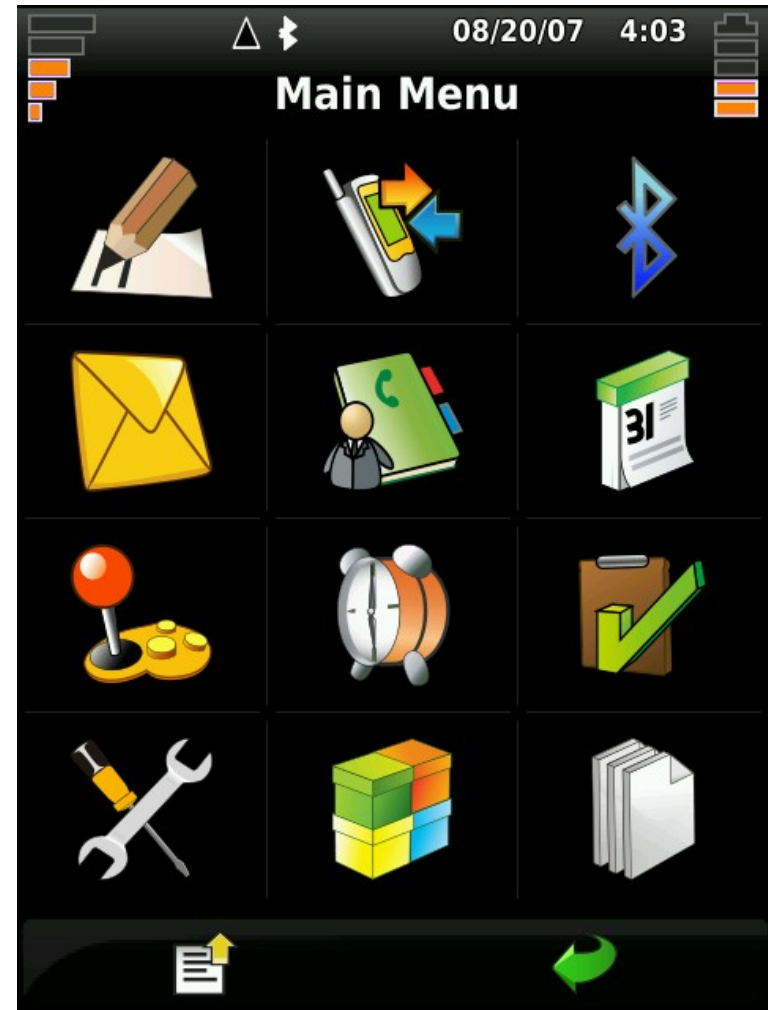
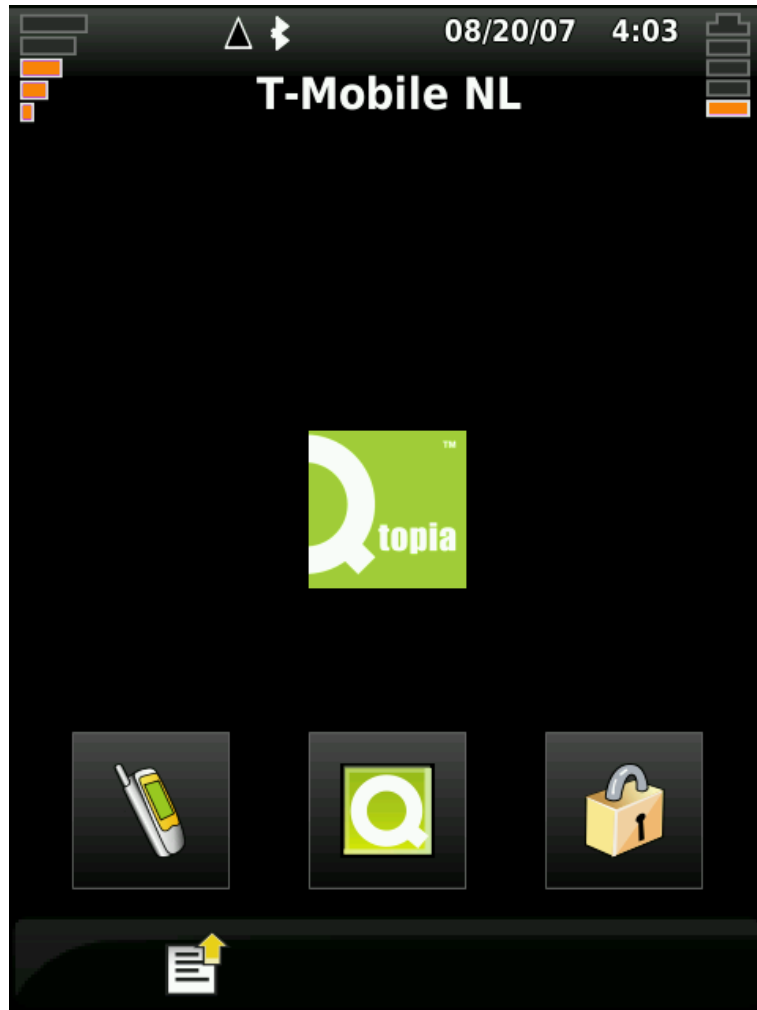
# Q-topia

Openmoko, come distribuzione, è ancora ad uno stadio embrionale.  
Sono riuscito a fare una chiamata solo attraverso i comandi AT!!!

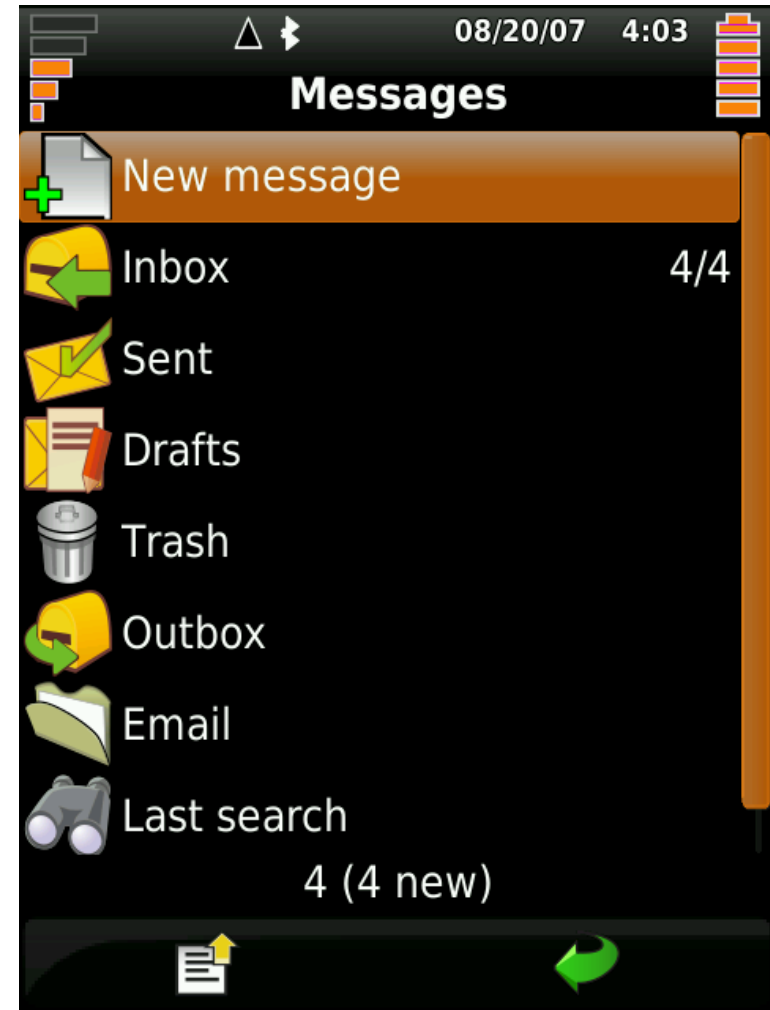
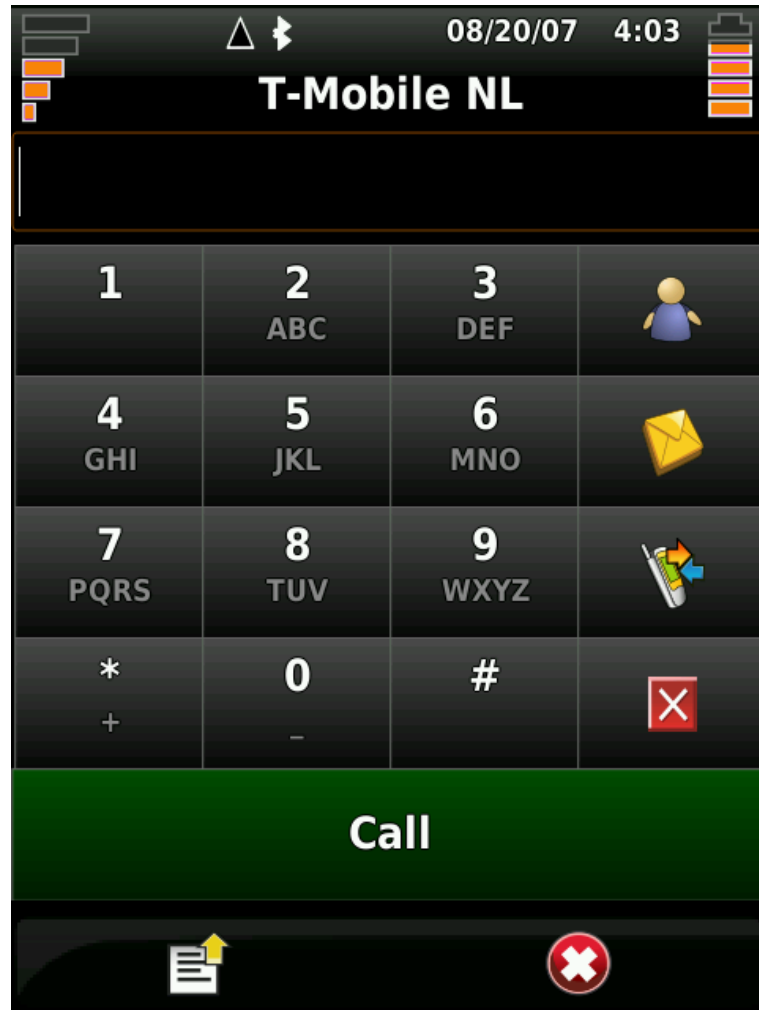
Ma ecco che ci viene in aiuto Q-topia con la sua distribuzione per il  
Greenphone.



# Alcuni screenshot: Main menu



# Alcuni screenshot: Dialer & messages



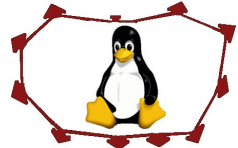
# Conclusioni

## PRO:

- L'hardware sembra ben supportato (u-boot è molto ben fatto e il kernel gestisce bene i vari dispositivi).
- Il sistema si presta bene per capire come funziona un sistema GNU/Linux embedded.
- Si ha la possibilità di toccare con mano (direttamente dalla command line) le impostazioni dei vari dispositivi.
- Il fatto di avere anche Q-topia «abbastanza» funzionante permette di usare il telefonino «normalmente».

## CONTRO:

- Openmoko è ad uno stato molto immaturo e i tempi di sviluppo sembrano molto lunghi...
- L'alimentazione via USB non sembra adatta ad un dispositivo tipo cellulare dove (per ora) gli assorbimenti rimangono alti.



# Riferimenti

<http://ftp.enneenne.com/pub/docs/talks/openmoko>

<http://www.openmoko.org>

<http://wiki.openmoko.org>

[http://wiki.openmoko.org/wiki/Qtopia\\_on\\_Neo\\_1973](http://wiki.openmoko.org/wiki/Qtopia_on_Neo_1973)

