

# Rapporti tecnici

# INGV

**Site Show Roma2:  
sistema di visualizzazione siti web**

# 324



## **Direttore Responsabile**

Stefano GRESTA

## **Editorial Board**

Luigi CUCCI - Editor in Chief (INGV - RM1)

Raffaele AZZARO (INGV-CT)

Mario CASTELLANO (INGV-NA)

Viviana CASTELLI (INGV-BO)

Rosa Anna CORSARO (INGV-CT)

Mauro DI VITO (INGV-NA)

Antonio GUARNIERI (INGV-BO)

Marcello LIOTTA (INGV-PA)

Mario MATTIA (INGV-CT)

Milena MORETTI (INGV-CNT)

Nicola PAGLIUCA (INGV-RM1)

Umberto SCIACCA (INGV-RM2)

Alessandro SETTIMI (INGV-RM2)

Salvatore STRAMONDO (INGV-CNT)

Andrea TERTULLIANI (INGV-RM1)

Aldo WINKLER (INGV-RM2)

## **Segreteria di Redazione**

Francesca Di Stefano - Referente

Rossella Celi

Tel. +39 06 51860068

redazionecen@ingv.it

in collaborazione con:

Barbara Angioni (RM1)



# Rapporti tecnici INGV

## **SITE SHOW ROMA2: SISTEMA DI VISUALIZZAZIONE SITI WEB**

Anna Grazia Buono, Adriano Azzarone e Achille Zirizzotti

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Geomagnetismo, Aeronomia e Geofisica Ambientale)

# 324



## Sommario

Introduzione.....	7
1. Il sistema di visualizzazione.....	7
1.1. Il Raspberry Pi.....	8
2. Sistema operativo.....	10
2.1 Configurazione del sistema operativo.....	10
2.2 Aggiornamenti.....	12
2.3 Ambiente grafico LXDE.....	12
2.4 Amministrazione e configurazione.....	13
2.5. Configurazione della rete Wi-Fi.....	15
3. Programmi per la visualizzazione automatica dei siti.....	16
3.1 Lo <i>script</i> per PhantomJS.....	17
3.2 Avvio automatico dello <i>script</i> .....	18
3.3 Visualizzazione delle immagini.....	19
3.4 Visualizzazione automatica delle immagini.....	19
4. Conclusioni.....	20
Bibliografia.....	20
Sitografia.....	20
Appendice A.....	23



## Introduzione

Il controllo regolare degli osservatori e delle reti geofisiche gestite dalla Sezione Roma2 viene effettuato attraverso alcune pagine web dedicate, in particolare quelle sui dati geofisici acquisiti e quelle con la descrizione delle attività di servizio della Sezione Roma2. La rappresentazione grafica dei dati acquisiti permette anche di monitorare il funzionamento delle varie strumentazioni di acquisizione. Per pubblicizzare le attività svolte è stata proposta l'idea, dalla Unità Funzionale Laboratori per l'Innovazione Tecnologica, di installare un monitor di grande dimensione in un luogo di passaggio dove poter mostrare in successione le suddette pagine web. L'idea è stata approvata dal Direttore di Sezione, dopo di che è stato realizzato un prototipo posizionato (temporaneamente) presso l'ingresso del corridoio della sede romana della Sezione (Figura 1).

Lo scopo di questo rapporto tecnico è quello di descrivere il sistema di visualizzazione "Site Show Roma2" con i dispositivi elettronici e i programmi utilizzati, ma anche di fornire un utile strumento di consultazione per il mantenimento o l'aggiornamento futuro del sistema.

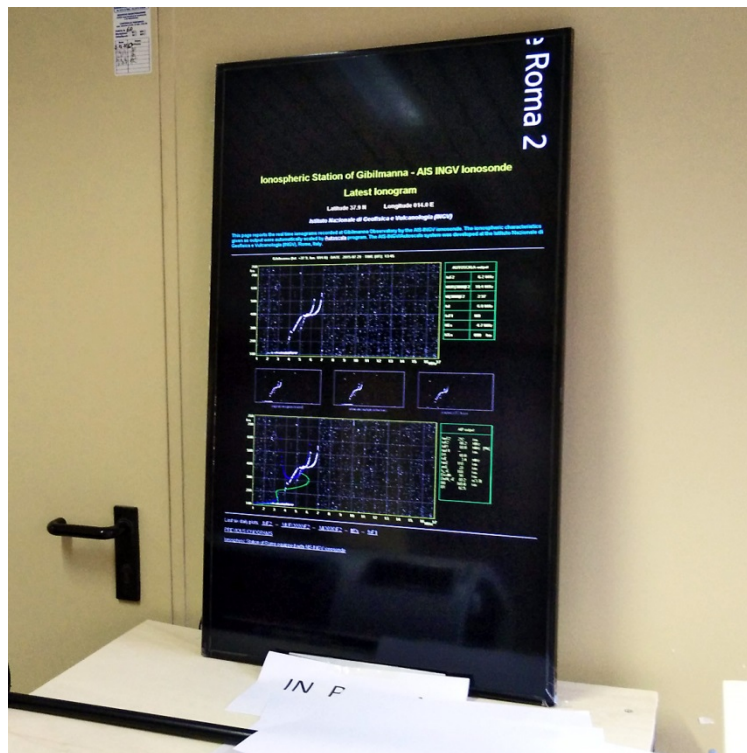


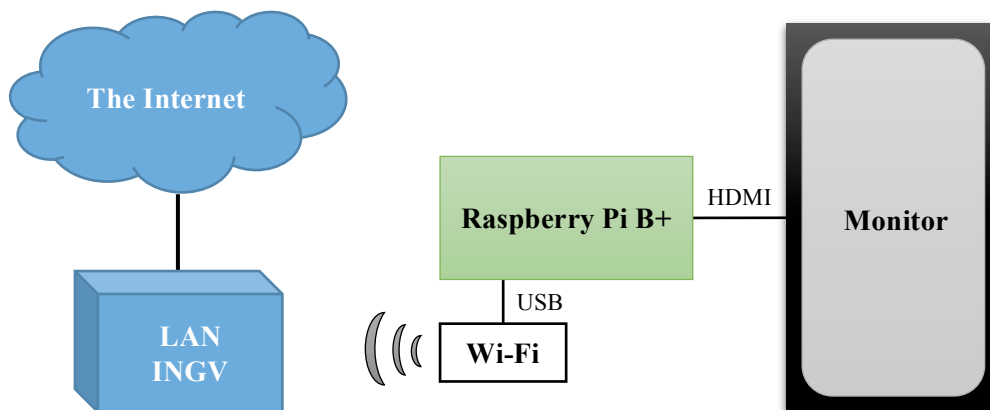
Figura 1. Prototipo sistema "Site Show Roma2".

### 1. Il sistema di visualizzazione

La visualizzazione di pagine web su monitor è un problema apparentemente banale, ma le pagine hanno sempre dimensioni e rapporto altezza-larghezza variabili tra loro, rendendo complicato adattarele a una superficie fissa di visualizzazione. Il problema delle differenti dimensioni in *pixel* delle pagine web è stato risolto attraverso una trasformazione in formato immagine, più facile da adattare e gestire. Il monitor è stato orientato verticalmente, come visibile in Figura 1, poiché generalmente le pagine web hanno lunghezza maggiore della larghezza.

Il sistema di visualizzazione, illustrato schematicamente in Figura 2, è composto da un monitor Samsung da 40 pollici modello LH40DB, appositamente progettato per la visualizzazione di immagini pubblicitarie, collegato ad un piccolo computer che svolge le funzioni di acquisizione delle pagine web, conversione in *files* immagine e visualizzazione delle stesse. Il dispositivo utilizzato è un **Raspberry Pi B+** [<http://www.raspberrypi.org>], un piccolo computer su singola scheda [Calderan, 2013]. Nonostante le ridotte dimensioni ed un costo contenuto (circa 20 euro), ha caratteristiche *hardware* che gli permettono di utilizzare

diversi sistemi operativi Linux a scelta dell'utente. Inoltre, ha la possibilità di interagire con altri componenti elettrici/elettronici per mezzo di interfacce tra cui citiamo GPIO (General Purpose Input/Output), porte USB e di rete. Il computer è collegato alla rete Wi-Fi di Istituto tramite un adattatore Wi-Fi USB, permettendo di posizionare il monitor in una zona svincolata dalla presa di rete Ethernet, che non sempre è presente nei punti di passaggio, quali corridoi o ingressi degli edifici, che invece sono posizioni ottimali per la presenza di un monitor.



**Figura 2.** Schema della configurazione utilizzata.

La trasformazione delle pagine web in immagini è eseguita tramite **PhantomJS**, un programma che permette di ricavare una stampa in formato immagine delle pagine web di nostro interesse (vedi paragrafo 3.1). Le immagini vengono mostrate a pieno schermo e in successione utilizzando **Feh**, un leggero e potente visualizzatore di immagini (vedi paragrafo 3.3).

### 1.1. Il Raspberry Pi

L'idea nasce nel 2006 da un gruppo di ricercatori dell'Università di Cambridge, preoccupati dal calo di competenze informatiche di bambini e giovani studenti, legato agli alti costi dei personal computer [McManus, 2013], ma anche al fatto che l'uso costante di sistemi operativi evoluti e ricchi di funzioni e interfacce grafiche non stimola ad approfondire gli ambienti di programmazione. L'idea si concretizza nel febbraio 2012, quando Raspberry Pi viene presentato in due modelli ("A" e "B"), al prezzo contenuto di 25 e 35 dollari, rispettivamente per la versione con 256 MB di RAM, una porta USB e senza porta Ethernet e la versione con 512 MB di RAM, due porte USB e porta Ethernet.

L'origine del nome riprende una tradizione britannica nell'utilizzo di nomi di frutta negli ambienti informatici ("raspberry" in inglese significa lampone), tra cui citiamo la famosa azienda Apple (mela), ma anche le aziende Acorn Computer (ghianda), Apricot (albicocca) e Tangerine (mandarino). La sigla "Pi" invece, fa riferimento al linguaggio di programmazione *Python* [<http://www.python.org/>], adottato come linguaggio di programmazione principale.

Il computer [Computing at School, 2012], ha dimensione compatta a "carta di credito" con varie porte di *input/output* come visibili in Figura 3 e con caratteristiche tecniche elencate in Tabella 1.

Il Raspberry Pi utilizza un processore di tipo **ARM** (Advanced RISC Machine), molto comune nei dispositivi mobili come smartphone o tablet. La sigla **RISC** (Reduced Instruction Set Computing) indica una caratteristica di progettazione del processore che integra un set di istruzioni "ridotto" che si contrappone al tipo **CISC** (Complex Instruction Set Computing) che integra un numero maggiore ("complesso") di istruzioni. Entrambe le tipologie di progettazione hanno vantaggi e svantaggi [Eric Roberts, 2000], ma quella RISC ha una maggiore efficienza energetica. La maggior parte dei computer oggi esistenti (portatili, desktop o server), utilizza processori di tipo CISC prodotti da Intel o AMD.

Il motivo principale per cui i processori con architettura ARM hanno avuto tanto successo nel mercato mobile è proprio dovuto al consumo di energia contenuto rispetto ad altre architetture. Visto che le batterie non sono migliorate molto negli ultimi anni, è rimasta forte la spinta tecnologica verso l'ottimizzazione energetica per estendere la durata della batteria.



Il Raspberry Pi modello B+ consuma circa 600mA se non sono collegate periferiche USB, altrimenti si può arrivare sino a 1,8A, per cui è consigliato usare un alimentatore a 2A; in un utilizzo non intensivo può essere comunque sufficiente un più comune alimentatore da 1° [<https://www.raspberrypi.org/help/faqs/#topPower>].

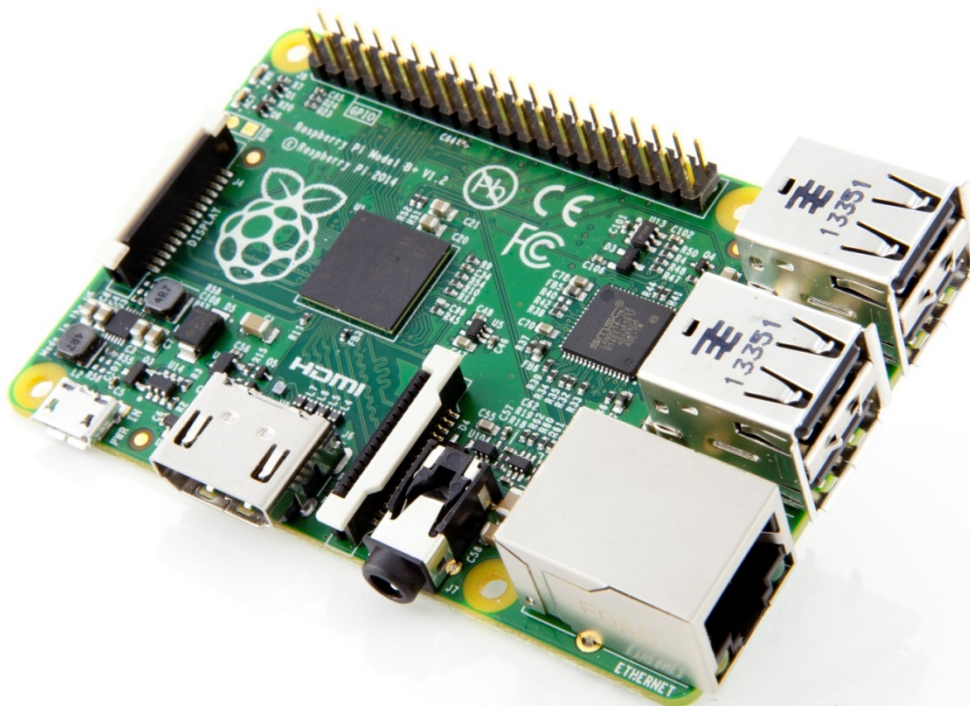
Lo scarso consumo di corrente ha come ulteriore vantaggio una bassa produzione di calore, pertanto la scheda non richiede generalmente un sistema di raffreddamento con dissipatori o ventole; in certe circostanze però, di utilizzo intenso o temperature ambientali elevate, è comunque consigliato applicare un dissipatore metallico sul processore.

Raspberry Pi non dispone di controller per hard disk, ma prevede come supporto di memorizzazione una memoria flash su uno *slot* dedicato. La memoria utilizzata è una scheda SD (Secure Digital) nelle versioni precedenti del Raspberry Pi e MicroSD in questa versione.

Il sistema operativo viene installato sulla memoria attraverso un *file* immagine disco che viene copiato con un programma adatto utilizzando un computer tradizionale (vedi paragrafo 2).

La connessione di rete Ethernet è di facile configurazione tramite il sistema operativo, grazie anche alla disponibilità dei protocolli DHCP (Dynamic Host Configuration Protocol), per ricevere automaticamente dalla rete un indirizzo IP, e Auto MDI-X (Auto Medium Dependent Interface Crossover), che elimina la necessità di utilizzare un cavo Ethernet “incrociato” altrimenti necessario per connettere assieme due dispositivi di rete dello stesso livello.

Inoltre, il collegamento alla rete può essere fatto anche tramite un comune adattatore Wi-Fi USB da acquistare separatamente; raramente si osservano incompatibilità, ma è comunque presente online un elenco di dispositivi testati [[http://elinux.org/RPi\\_USB\\_Wi-Fi\\_Adapters](http://elinux.org/RPi_USB_Wi-Fi_Adapters)].



**Figura 3.** Foto di Raspberry Pi modello B+. Connettori partendo dall’alto in senso orario: GPIO 40 pin; 4 porte USB 2.0; porta Rj45 di rete Ethernet 10/100; connettore CSI (Camera Serial Interface) per collegare una videocamera; presa audio e video compositi integrati in porta di tipo Jack a 4 poli; porta HDMI con output video e audio fino a risoluzione 1080p; porta microUSB per alimentazione a 5V; connettore DSI (Serial Digital Interface) interfaccia video (al momento non sono disponibili dispositivi compatibili).

<b>Specifiche</b>	
<b>Chip</b>	Broadcom BCM2835 SoC
<b>Core architecture</b>	ARM11
<b>CPU</b>	700 MHz Low Power ARM1176JZFS Applications Processor
<b>GPU</b>	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure.
<b>Memory</b>	512MB SDRAM
<b>Sistema Operativo</b>	Boots from MicroSD card, running a version of the Linux operating system
<b>Dimensioni</b>	85 x 56 x 17mm
<b>Alimentazione</b>	Micro USB socket 5V, 600mA to 1.8A max
<b>Connessioni</b>	
<b>Ethernet</b>	10/100 BaseT Ethernet socket
<b>Uscita Video</b>	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
<b>Uscita Audio</b>	3.5mm jack, HDMI
<b>USB</b>	4x USB 2.0
<b>GPIO</b>	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip. Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines.
<b>Camera</b>	15-pin MIPI Camera Serial Interface (CSI-2)
<b>JTAG</b>	Not populated
<b>Display</b>	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
<b>Memory Card Slot</b>	MicroSD SDXC

**Tabella 1.** Caratteristiche tecniche di Raspberry Pi modello B+.

## 2. Sistema operativo

L'architettura ARM non è attualmente compatibile con i tradizionali sistemi operativi proprietari (Microsoft Windows o Apple Mac OS), anche se il nuovo "Windows 10" supporterà ARM e Microsoft ha certificato la compatibilità con la seconda versione del Raspberry [<https://www.raspberrypi.org/blog/windows-10-for-iot/>].

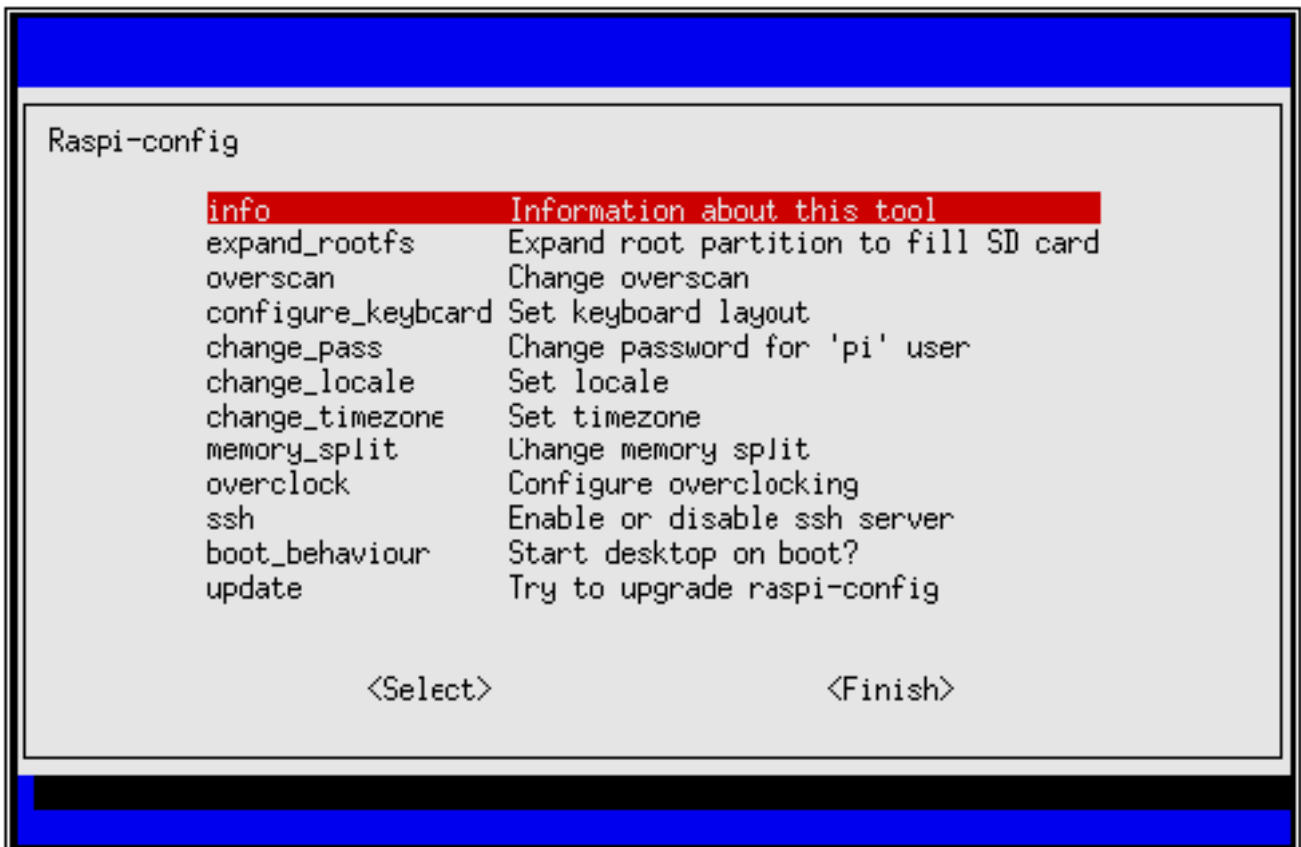
Attualmente sul Raspberry Pi utilizziamo un sistema operativo basato su GNU/Linux, *open source* e consigliato dal produttore: "Raspbian", che offre più di 35.000 pacchetti *software* precompilati disponibili [<https://www.raspbian.org/>]. Tutte i sistemi operativi sono sempre forniti come un singolo *file* immagine disco da scaricare e copiare nella memoria MicroSD attraverso specifici programmi.

### 2.1 Configurazione del sistema operativo

L'installazione del sistema operativo "Raspbian" è stata fatta su una scheda di memoria MicroSD Card Sandisk da 16 GB formattata con *filesystem* FAT32. Una volta scaricato *online* il *file* immagine disco del sistema operativo, si è provveduto a copiarlo sulla memoria con il programma "Win32DiskImager" (suggerito per i sistemi Windows), il cui utilizzo è abbastanza immediato e l'operazione richiede circa un'ora [<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>]. Con il programma è anche possibile creare un *file* immagine disco del sistema operativo presente nella MicroSD come copia di backup aggiornata con le ultime configurazioni effettuate; infatti, in caso di interruzione della alimentazione, soprattutto durante una operazione di scrittura sulla memoria, potrebbe danneggiarsi il contenuto memorizzato.

Terminate queste operazioni si inserisce la scheda nel Raspberry Pi e, dopo l'accensione, viene visualizzata la schermata di configurazione "Raspi-config" come in Figura 4.

Già dal primo avvio del dispositivo sono presenti tutti i *driver* essenziali, in particolare per le periferiche USB di tastiera e mouse necessarie a modificare le impostazioni di configurazione.



**Figura 4.** Schermata del programma di configurazione iniziale.

Il programma di configurazione propone un menù con diverse informazioni e impostazioni:

- “info”, contiene le informazioni sul *tool* di configurazione.
- “expand\_rootfs”, permette di estendere l’immagine del sistema operativo su tutta la dimensione della scheda di memoria, visto che durante la preparazione della stessa il *file* immagine va ad occupare solamente lo spazio dell’immagine di origine (circa 2GB); questa estensione verrà effettuata al successivo riavvio del dispositivo.
- “overscan”, permette di modificare i valori per adattare l’immagine allo schermo.
- “configure\_keyboard”, permette di impostare il *layout* della tastiera USB collegata.
- “change\_pass”, permette di cambiare la password dell’utente “pi”.
- “change\_locale”, imposta il linguaggio locale usato nel *set* di caratteri.
- “change\_timezone”, imposta l’ora in base al fuso orario locale con la possibilità di aggiornare automaticamente l’orario se connessi ad internet.
- “memory\_split”, cambia l’assegnazione della memoria dedicata al VideoCore e al processore. È consigliabile lasciare l’impostazione di default, che assegna 64 Mb al VideoCore e 192 Mb al processore ARM.
- “overclock”, permette di configurare l’*overclock* della CPU. È consigliabile lasciare le impostazioni di default.
- “ssh”, permette di abilitare/disabilitare il protocollo di rete SSH (Secure Shell) del *server* permettendo di accedere al Raspberry Pi dalla rete. È consigliato attivarlo.
- “boot\_behaviour”: imposta l’avvio automatico del desktop al *boot*, ossia viene lanciata in automatico la modalità grafica all’avvio del dispositivo. Se non è impostato, la modalità grafica deve essere lanciata manualmente tramite il comando *startx*.
- “update”: aggiorna il sistema operativo all’ultima versione.

## 2.2 Aggiornamenti

Il programma “apt” (Advanced Packaging Tools) è in grado di scaricare pacchetti *software* contemporaneamente da diversi sorgenti collegate fra loro (*repository*) per aggiornare il sistema operativo o installare programmi. È consigliabile aggiornare periodicamente i pacchetti *software* installati con il comando `sudo apt-get update`. L’installazione di un pacchetto si effettua con il comando `sudo apt-get install <nome_pacchetto>`, mentre per disinstallarlo si usa il comando `sudo apt-get remove <nome_pacchetto>`. Infine per aggiornare nuove versioni dei pacchetti installati nel sistema si usa il comando `sudo apt-get upgrade`.

L’aggiornamento del *kernel* del sistema operativo è possibile attraverso una sequenza di operazioni:

1. installare “git” (sistema di controllo versione);  
`sudo apt-get install git-core`
2. scaricare gli ultimi aggiornamenti;  
`sudo apt-get update && sudo apt-get dist-upgrade`
3. aggiornare il *kernel*;  
`sudo wget http://goo.gl/1BOfJ -O /usr/bin/rpi-update && chmod +x /usr/bin/rpi-update`
4. aggiornare il *firmware*;  
`sudo rpi-update`
5. verificare la versione di sistema installato (Figura 5);  
`uname -a`
6. verificare la versione del *firmware* (Figura 6);  
`/opt/vc/bin/vcgencmd version`
7. riavviare il sistema.  
`shutdown -r now`

```
pi@raspberrypi ~ $ uname -a
Linux raspberrypi 3.18.5+ #746 PREEMPT Mon Feb 2 13:57:16 GMT 2015 armv6l GNU/Linux
```

Figura 5. Controllo versione di sistema.

```
pi@raspberrypi ~ $ /opt/vc/bin/vcgencmd version
Jan 30 2015 18:25:11
Copyright (c) 2012 Broadcom
version d6e004c61a7a749897c482c860d0b2c28196437e (clean) (release)
```

Figura 6. Verifica versione del *firmware*.

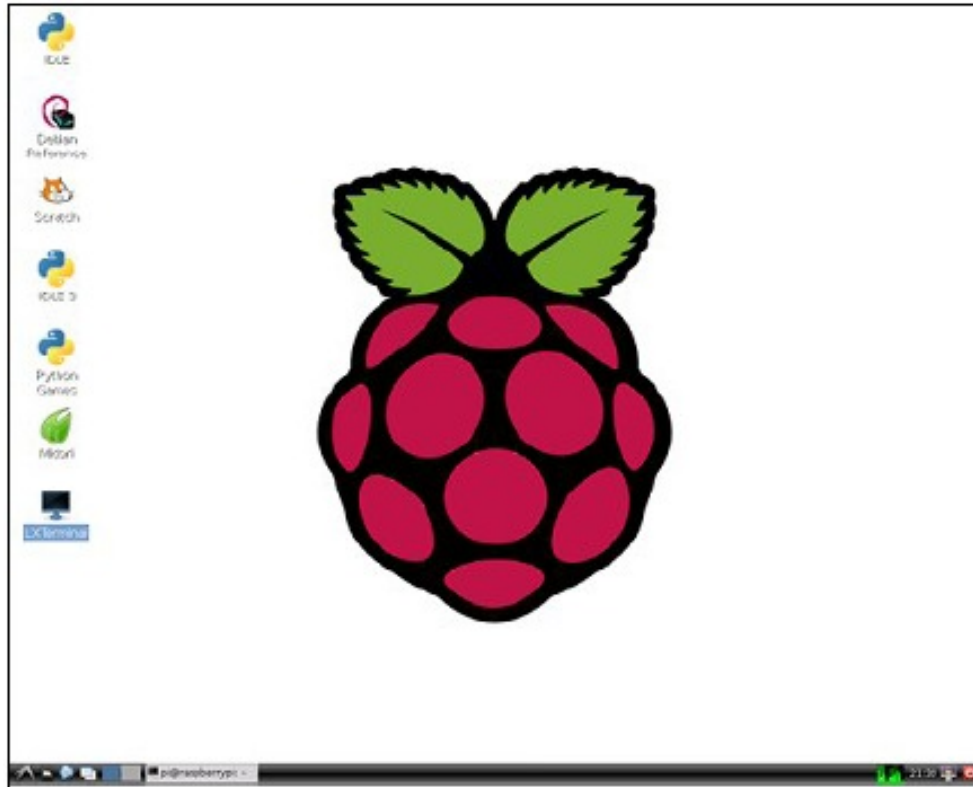
## 2.3 Ambiente grafico LXDE

LXDE (Lightweight X11 Desktop Environment) è un ambiente grafico gratuito per GNU/Linux, creato con l’obiettivo di essere leggero in termini di memoria occupata. LXDE è veloce, performante, consuma poche risorse *hardware* ed è mantenuto da una comunità internazionale di sviluppatori. Si presenta con una interfaccia semplice, supporto multi-lingua, tasti di scelta rapida standard e funzioni aggiuntive. Utilizza meno processore e RAM di altri ambienti grafici, infatti, per quanto riguarda la RAM ad esempio, LXDE ne occupa 98 MB, mentre un altro ambiente come LXQt v0.9 ne occupa 118 MB e XFCE ne occupa 107 MB.

L’ambiente grafico si avvia automaticamente all’accensione del Raspberry Pi (se impostato con “raspi-config” o se modificato il *file* di configurazione “/etc/xdg/lxsession/LXDE/autostart”, vedi paragrafo 3.4). Vengono subito visualizzati sul *desktop* (Figura 7) una serie di programmi preinstallati, tra questi notiamo:

- “Midori”, un *web browser* ottimizzato in modo da richiedere poche risorse alla CPU;
- un programma di compilazione per il linguaggio di programmazione *Python*;
- “WifiConfig”, che permette la configurazione della rete Wi-Fi;
- “LXTerminal”, emulatore di terminale che permette di eseguire comandi testuali.

Per modificare le impostazioni grafiche e ottimizzare la visualizzazione sullo schermo in uso occorre apportare delle modifiche al *file* di configurazione “config.txt”. Per far questo occorrono dei privilegi di amministratore che si possono ottenere (vedi paragrafo successivo) digitando la seguente istruzione `sudo leafpad /boot/config.txt`, dalla voce “run” aprendo il menù di sistema (prima icona a sinistra sulla barra applicazioni in basso) o tramite “LXTerminal”.



**Figura 7.** Ambiente grafico LXDE.

## 2.4 Amministrazione e configurazione

I sistemi operativi basati su Linux utilizzano il termine “superuser” a indicare il livello di privilegio massimo dell’amministratore di sistema (il cui nome utente di default è “root”). Il passaggio al livello “superuser” avviene con il comando `su` e l’inserimento della password di “root”. Nel caso in cui non si abbia una buona conoscenza dell’ambiente Linux, essere “superuser” può essere pericoloso in quanto si ha il potere di eseguire ogni operazione sul sistema anche quelle potenzialmente dannose, per cui conviene rimanere al livello dell’utente “pi” e utilizzare il prefisso `sudo` (acronimo di “super user do”) nei comandi ogni volta che si vuole agire da “superuser”, ad esempio quando si vuole installare programmi o aggiornare la configurazione del sistema (la schermata di configurazione iniziale andrà richiamata digitando il comando: `sudo raspi-config`).

Per spegnere il Raspberry Pi occorre digitare il comando `sudo shutdown -h now` (o `sudo halt`), dove “-h” spegne il dispositivo dopo lo “shutdown” del sistema operativo, mentre “now” esegue il comando istantaneamente. Oppure si può specificare un numero che corrisponde ai minuti di attesa prima che il comando sia eseguito. Il riavvio del sistema si effettua con il comando `sudo shutdown -r now` oppure `sudo reboot`.

Per modificare un *file* di testo o di configurazione (per cambiare delle impostazioni di sistema) si possono utilizzare diversi programmi, tra cui “nano” dal *prompt* dei comandi oppure “Leafpad” dal menù applicazioni o cliccando con il pulsante destro sul *file* da modificare.

Seguono una serie di comandi utili alla verifica del funzionamento del Raspberry Pi:

- `free`  
Verifica della quantità di memoria utilizzata e quella disponibile (Figura 8).

- `df /`  
Verifica della quantità di spazio utilizzato su disco (Figura 9).
- `/opt/vc/bin/vcgenclmd measure_temp`  
Controllo della temperatura processore. In Figura 10 la temperatura presenta un valore accettabile, ma se si superasse il valore di 80°C potrebbero nascere dei malfunzionamenti.
- `hostname -I`  
Verifica del collegamento in rete dell'indirizzo IP assegnato alla scheda di rete (Figura 11).
- `Ifconfig`  
Visualizza i dettagli delle schede di rete del dispositivo e la loro configurazione [Schroder, 2007] (Figura 12).
- `chmod`  
Modifica i permessi di *files* e *directory*. In Tabella 2 sono visualizzati i codici assegnati ad ogni tipo di permesso. La sintassi del comando è `chmod UGO <nome file>`, dove “U” rappresenta il livello di permessi utente, “G” il livello di permessi di gruppo e “O” il livello di permessi generale. Ad esempio per rendere eseguibile all'utente il *file batch* “slide.sh” si utilizza l'istruzione `sudo chmod 755 slide.sh`.

```
pi@raspberrypi ~ $ free
              total        used         free       shared    buffers     cached
Mem:           380716      182776      197940           0         20892       81452
-/+ buffers/cache:
Swap:          102396          116       102280
```

Figura 8. Memoria utilizzata.

```
pi@raspberrypi ~ $ df /
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root        15004680 3133200  11225740  22% /
```

Figura 9. Spazio sul disco (memoria SD).

```
pi@raspberrypi ~ $ /opt/vc/bin/vcgenclmd measure_temp
temp=48.7'C
```

Figura 10. Temperatura.

```
pi@raspberrypi ~ $ hostname -I
10.180.3.17
```

Figura 11. Indirizzo IP.

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:96:68:46
          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)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  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)

wlan0     Link encap:Ethernet  HWaddr 80:1f:02:9b:b6:ba
          inet addr:10.180.3.17  Bcast:10.183.255.255  Mask:255.252.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:133567 errors:0 dropped:10347 overruns:0 frame:0
          TX packets:81087 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:101639161 (96.9 MiB)  TX bytes:11149370 (10.6 MiB)
```

Figura 12. Informazioni sulle schede di rete presenti.

Valore binario (rwx)	Valore decimale	Permessi
111	7	lettura, scrittura ed esecuzione
110	6	lettura e scrittura
101	5	lettura ed esecuzione
100	4	solo lettura
011	3	scrittura ed esecuzione
010	2	solo scrittura
001	1	solo esecuzione
000	0	Nessuno

**Tabella 2.** Tabella dei permessi.

## 2.5. Configurazione della rete Wi-Fi

Per abilitare la rete Wi-Fi in modalità automatica si deve modificare il *file* “interfaces” con il comando `sudo nano /etc/network/interfaces` ed effettuare le seguenti modifiche:

1. `auto wlan0`  
attiva il dispositivo all'accensione;
2. `iface wlan0 inet dhcp`  
imposta “wlan0” in modalità DHCP;
3. `wireless-essid pi`  
asigna il nome della rete *wireless* utilizzata.

Come mostrato nella Figura 13 il *file* “interfaces” contiene al suo interno le possibili opzioni commentate, per una più facile configurazione. Al termine occorre salvare le modifiche fatte e riavviare il sistema.

```

GNU nano 2.2.6      File: /etc/network/interfaces
auto lo

iface lo inet loopback
iface eth0 inet dhcp
#static
#iface eth0 inet static
#address 192.168.58.160
#netmask 255.255.255.0
#gateway 192.168.58.160

allow-hotplug wlan0
auto wlan0
iface wlan0 inet dhcp
wireless-essid pi
pre-up wpa_supplicant -B w -D wext -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.conf
#post-down killall -q wpa_supplicant
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

```

**Figura 13.** Impostazioni rete Wi-Fi.

Per rendere più compatto il sistema monitor & Raspberry Pi, l'installazione sarà fatta senza collegare tastiera e mouse. Modifiche alla configurazione e operazioni di manutenzione possono essere svolti utilizzando un terminale remoto via LAN o Wi-Fi con protocollo di rete SSH (Secure Shell), ad esempio usando l'applicazione “PuTTY” per i sistemi Windows, inserendo l'indirizzo IP associato (come in Figura 14); cliccando sul bottone “Open” si aprirà una finestra dove verranno chieste le credenziali di autenticazione e successivamente si potranno eseguire operazioni da remoto via terminale sul Raspberry Pi.



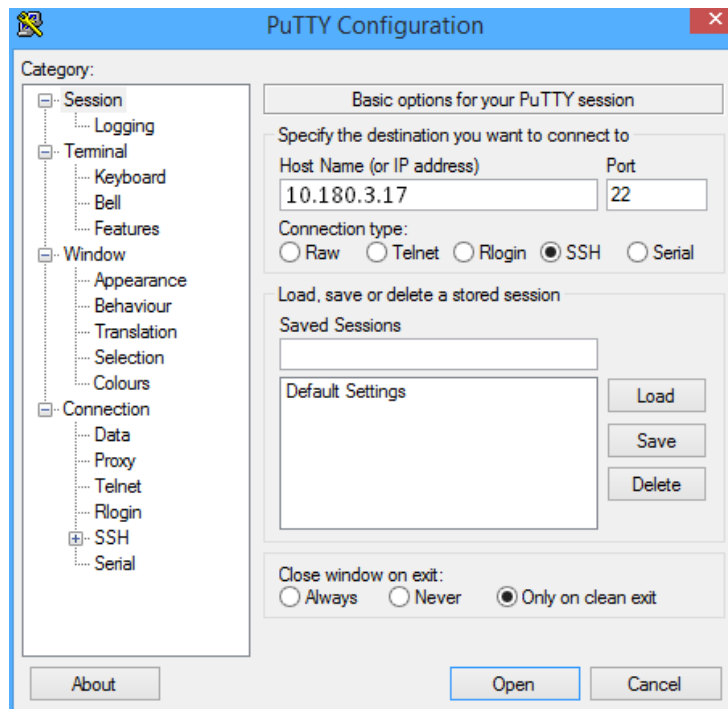


Figura 14. Schermata di configurazione PuTTY.

### 3. Programmi per la visualizzazione automatica dei siti

Per generare le immagini dei siti web viene usato il programma **PhantomJS** (tramite il *file script* “roma2\_sitshow.js”), mentre per la visualizzazione dei *files* immagine il programma **Feh**. Il formato dei *files* generati è PNG (Portable Network Graphics) a 32 bit, capace di memorizzare immagini in modo *lossless* (senza perdita), ossia senza perdere alcuna informazione su colori e qualità.

Per visualizzare i siti web sarebbe stato possibile utilizzare un normale *browser* (ad esempio Firefox) e una pagina web appositamente costruita che automaticamente aprisse uno dopo l’altro i vari siti web. Questa soluzione è stata testata, ma ha evidenziato alcuni problemi:

- lentezza nell’apertura dei siti, dovuta alla piattaforma *hardware* utilizzata, poco potente;
- problemi di visualizzazione di alcuni siti, ad esempio quando la risoluzione del monitor è inferiore alla reale dimensione del sito, risulta che una parte del sito non venga mostrata;
- possibili problemi nel caso di interruzioni di rete o internet, per cui il *browser* non avrebbe potuto aprire le pagine mostrando invece una pagina di errore.

Si è quindi cercata una soluzione alternativa, in cui vengano visualizzate le immagini dei siti web, precedentemente salvate, come in uno *slideshow*, con il risultato di avere un sistema più veloce, fedele nella visualizzazione e più affidabile in caso di problemi di connettività internet o di rete.

Per prima cosa, occorre impostare l’avvio del sistema senza richiesta di login, apportando delle modifiche al *file* “/etc/rc.local” e inserendo la stringa di comando su `-l pi -c startx` come illustrato in Figura 15.



```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

su -l pi -c startx &
#sudo xinit ./home/pi/startMidor &
exit 0
```

Figura 15. Eliminazione login all’accesso.

Come detto, per avere una visualizzazione ottimale dei siti web, è preferibile orientare il monitor in posizione verticale; per ruotare di conseguenza anche l’interfaccia grafica si deve modificare il *file* di configurazione “/boot/config.txt”, togliendo il commento “#” alla riga “display\_rotate=3”, come mostrato in Figura 16.

```
# uncomment to increase signal to HDMI, if you have interference, blanking,
#display_rotate=0
#display_rotate=1
#display_rotate=2
display_rotate=3
#display_rotate=0x10000
#display_rotate=0x20000
```

Figura 16. Parametri rotazione schermo del *file* “/boot/config.txt”.

### 3.1 Lo script per PhantomJS

**PhantomJS** [<http://phantomjs.org/>] è un *headless browser*, cioè un *browser* senza interfaccia grafica, che integra il motore di *rendering* di pagine web “WebKit”. Il programma è configurabile tramite *file script* in linguaggio *Javascript* per eseguire tutta una serie di operazioni quali testing, interazione, monitoraggio o salvataggio in immagini di pagine web (come nel nostro caso). È un software *open source*, multipiattaforma, rilasciato sotto licenza BSD (Berkeley Software Distribution) e disponibile anche in una versione specifica per il Raspberry Pi (“Phantomjs-linux-armv6l”).

Per l’installazione occorre eseguire le seguenti istruzioni:

1. `cd /tmp`  
entrare nella cartella contenete i *files* temporanei;
2. `wget https://github.com/aerberhardo/phantomjs-linux-armv6l/archive/master.zip`  
scaricare il *file* “master.zip”;
3. `unzip master.zip`  
scompattare il *file* “master.zip”;
4. `cd phantomjs-linux-armv6l-master`  
accedere alla *directory* già creata a seguito dell’estrazione dei *files* contenuti nel *file* zip;
5. `bunzip2 *.bz2 && tar xf *.tar`  
estrarre tutti i *files* contenuti nelle cartelle compresse;

6. `./phantomjs-1.9.0-linux-armv6l/bin/phantomjs -version 1.9.0`  
avviare l'installazione del compilatore (nota: la versione 1.9.0);
7. `sudo mv /home/pi/Desktop/ phantomjs-linux-armv6l-master /tmp/phantomjs-linux-armv6l-master/`  
spostare i *files* nella *directory* di lavoro.

Successivamente si è creata una cartella in “/home/pi/slideshow” (il contenuto è listato nella Figura 17) contenente lo *script* “roma2\_slideshow.js”, mandato in esecuzione con **PhantomJS** tutti i giorni ogni ora dalle 7 alle 19, in modo da avere un aggiornamento costante delle pagine web. I *files* immagine vengono creati nella stessa *directory* di lavoro. Lo *script* “roma2\_slideshow.js” è listato in Appendice A.

Le variabili importanti dello *script* sono:

- “arrayOfUrls”: *array* con la lista dei siti web da processare;
- “page.viewportSize”: indica la risoluzione dello “schermo virtuale” su cui le pagine web vengono “visualizzate” per essere salvate come immagini. La risoluzione usata è in accordo con lo standard FullHD (1080x1920px), che è anche la massima riproducibile dal Raspberry Pi tramite connessione HDMI.

```

pi@raspberrypi ~/slide $ ls
edita.sh          slideshow-11.png slideshow-17.png slideshow-6.png
phantomjs        slideshow-12.png slideshow-18.png slideshow-7.png
roma2_slideshow2.js slideshow-13.png slideshow-1.png  slideshow-8.png
roma2_slideshow3.js slideshow-14.png slideshow-2.png  slideshow-9.png
roma2_slideshow.js slideshow-15.png slideshow-4.png
slideshow-10.png slideshow-16.png slideshow-5.png

```

**Figura 17.** Cartella “slideshow” con il programma “phantomjs”, lo *script* “roma2\_slideshow.js” e i *files* immagine in formato PNG.

Nella *directory* “Desktop” è stato creato il *file batch* eseguibile “slide.sh”, contenente le poche istruzioni per iniziare la generazione delle immagini. Come mostrato in Figura 18, nella prima riga si cambia la *directory* e nella seconda si esegue **PhantomJS** con le impostazioni dello *script*.

```

cd /home/pi/slideshow/
./phantomjs roma2_slideshow.js

```

**Figura 18.** Contenuto del *file batch* “slide.sh”.

### 3.2 Avvio automatico dello script

Per aggiornare le immagini con gli ultimi dati acquisiti occorre eseguire periodicamente il *file batch* “/home/pi/Desktop/slide.sh” tramite **Crontab** [<http://crontab.org/>], componente dei sistemi operativi di tipo Unix che permette di pianificare una serie di operazioni a cadenza controllata temporalmente. Per cambiare le pianificazioni si usa il comando `crontab -e` che apre per la modifica il *file* di configurazione, ne verifica la correttezza e ne avvia l'utilizzo. La sintassi di una pianificazione è “\* \* \* \* \*/percorso/script.sh”, dove i caratteri asterisco “\*” rappresentano, in ordine: minuti (0-59); ore (0-23); giorno del mese (1-31); mese (1-12); giorno della settimana (0-7). L'operatore asterisco (\*) specifica tutti i possibili valori di un campo, ad esempio: un asterisco “\*” nel campo dell'ora è equivalente a «ogni ora». L'operatore trattino “-” specifica un intervallo di valori, ad esempio: “1-6” considera tutti i valori da 1 a 6. L'operatore virgola “,” specifica un raggruppamento di valori, ad esempio: “1-3, 5-9” considera sia valori da 1 a 3 che da 5 a 9. Per raggiungere un livello di personalizzazione più elevato è anche possibile utilizzare le espressioni regolari

Nel nostro caso abbiamo impostato **Crontab** con l'intervallo orario (al minuto 0 di ogni ora dalle 7 alle 19) nella riga “0 7-19 \* \* \* /home/pi/Desktop/slide.sh”, come visibile in Figura 19.

```
# m h dom mon dow command
0 7-19 * * * /home/pi/Desktop/slide.sh
```

Figura 19. Contenuto delle impostazioni per Crontab.

### 3.3 Visualizzazione delle immagini

Per visualizzare in modo automatico delle immagini si è deciso di utilizzare **Feh** [<http://feh.finalrewind.org/>], un programma leggero ma potente. Al contrario di altri programmi di visualizzazione, non dispone di una comoda interfaccia grafica, ma semplicemente mostra le immagini impostate. È controllabile attraverso riga di comando e configurabile con azioni e click del mouse sulle immagini visualizzate. Può essere anche utilizzato per la gestione degli sfondi del desktop, oltre ad essere configurabile per il funzionamento automatico. Per ottenere la lista completa di opzioni eseguire il comando `feh -help`.

Il comando `/usr/bin/feh -Z -F -Y -q -z -D 10 /home/pi/slide` esegue **Feh** con le opzioni di configurazione per impostare in modo automatico lo zoom, aprire le immagini a schermo pieno, attivare la modalità *fullscreen* e mostrare ogni immagine della *directory* “/home/pi/slide” per un tempo di 10 secondi.

Per installare **Feh** eseguire il comando `sudo apt-get install feh`.

### 3.4 Visualizzazione automatica delle immagini

Per rendere automatica la visualizzazione delle immagini tramite **Feh**, si è creato un *file batch* eseguibile denominato “start\_rend.sh” (Figura 20) contenente le istruzioni per:

- `xset s off`  
disabilitare lo screen saver;
- `xset -dpms`  
disabilitare il risparmio energetico;
- `xset s noblank`  
disabilitare autospegnimento dello schermo;
- `/usr/bin/feh -Z -F -Y -q -z -r -D 30 /home/pi/slideshow`  
eseguire l’applicazione Feh con le opzioni necessarie.

```
#!/bin/sh
xset s off #disable screen saver
xset -dpms #disable DPMS (Energy Star) features.
xset s noblank #dont'blank the video device
unclutter &
matchbox-windows-manager &
/usr/bin/feh -Z -F -Y -q -z -r -D 30 /home/pi/slideshow &
```

Figura 20. Contenuto del *file batch* “start\_rend.sh”.

Affinché questo *file batch* sia realmente eseguito all’avvio del Raspberry Pi, occorre fare delle modifiche nel *file* eseguibile “/etc/xdg/lxsession/LXDE/autostart”, come in Figura 21: `@lxpanel -profile LXDE`

permette di attivare un pannello per il “Desktop” nella modalità LXDE (vedi paragrafo 2.3);

- `@pcmanfs -desktop -profile LXDE`  
attiva un *file manager*, tra i più leggeri, impostato come predefinito da LXDE;
- `@xscreensaver -no-splash`  
“xscreenserver” è uno *screen server* appartenente alla famiglia degli *X Windows System*, per renderlo eseguibile bisogna attivarlo tramite il comando `@xscreensaver -no-splash`;
- `@sh /home/pi/start_feh.sh`  
Istruzione per mandare in esecuzione il *file* “start\_rend.sh”.

```
@lxpanel --profile LXDE
@pcmanfm --desktop --profile LXDE
@xscreensaver -no-splash
@sh /home/pi/start_rend.sh
```

**Figura 21.** Contenuto del file “/etc/xdg/lxsession/LXDE/autostart”.

## 4. Conclusioni

Questo documento illustra il metodo usato per creare un “Site Show Roma2”, ovvero un sistema che permette la visualizzazione in successione di alcune pagine web che riguardano principalmente grafici e tabelle dei dati acquisiti dagli Osservatori Geomagnetici e Ionosferici e dalle altre Reti di monitoraggio della Sezione Roma2. Il sistema è stato configurato e sarà installato presso l’ingresso del corridoio della sede romana della Sezione, dopo una iniziale fase di test con un prototipo, per pubblicizzare le attività della Sezione. Il sistema è composto da un monitor di grandi dimensioni e da un Raspberry Pi B+, un piccolo computer su singola scheda a basso costo, programmabile, con sistema operativo Linux. Tutti i programmi utilizzati e relative configurazioni sono stati descritti nei capitoli precedenti.

Il sistema presentato è versatile e di facile implementazione. Il costo è rappresentato soprattutto dal monitor scelto di grandi dimensioni, ma per altre installazioni si potrebbe usare un monitor più piccolo.

Il sistema si presta ad essere perfezionato e ad implementare nuove funzioni, come ad esempio creare un pannello di gestione online dove poter modificare la lista dei siti e altre opzioni di visualizzazione.

La seconda versione di Raspberry Pi, che dispone di un processore più potente pur mantenendo invariati i consumi energetici, sarà testata per verificarne la compatibilità e per essere usata eventualmente in nuove installazioni.

## Bibliografia

- Calderan P., (2013). *Raspberry Pi. Guida al computer più compatto del mondo*. Milano, Apogeo, ISBN: 9788850332243.
- McManus S. & Cook M. (2013). *Raspberry Pi For Dummies*. Hoboken, NJ, John Wiley & Sons. ISBN: 9781118554210.
- Computing at School (CAS) & Raspberry Pi Foundation. (2012). *The Raspberry Pi Education Manual*. URL: <https://www.raspberrypi.org/blog/cas-educational-manual/>
- Eric R., (2000). *RISC vs. CISC*. Stanford University. URL: <http://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/riscisc/>
- Schroder C., (2007). *Linux Networking Cookbook*. Sebastopol, CA, O’Reilly Media. ISBN: 9780596102487.

## Sitografia

- Sito ufficiale della Fondazione Raspberry Pi:  
<http://www.raspberrypi.org>
- Sito ufficiale del linguaggio di programmazione *Python*:  
<http://www.python.org/>
- Requisiti di alimentazione di Raspberry Pi:  
<https://www.raspberrypi.org/help/faqs/#topPower>
- Elenco di adattatori USB Wi-Fi testati compatibili con Raspberry Pi:  
[http://elinux.org/RPi\\_USB\\_Wi-Fi\\_Adapters](http://elinux.org/RPi_USB_Wi-Fi_Adapters)
- Annuncio compatibilità di Windows 10 per Raspberry Pi 2:  
<https://www.raspberrypi.org/blog/windows-10-for-iot/>
- Sistema operativo Raspbian:  
<https://www.raspbian.org/>
- Documentazione per installazione sistemi operativi su Raspberry Pi:

<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

Sito ufficiale del programma PhantomJS:

<http://phantomjs.org/>

Documentazione per Crontab:

<http://crontab.org/>

Sito ufficiale del programma Feh:

<http://feh.finalrewind.org/>

Adafruit, con guide e tutorial su progetti con Raspberry Pi:

<http://learn.adafruit.com/category/raspberry-pi>

Wiki sul Raspberry della community Linux:

[http://elinux.org/RPi\\_Hub](http://elinux.org/RPi_Hub)



# Appendice A





## Appendice A

*Script* “roma2\_sitshow.js”, in grassetto la parte che contiene la lista dei siti selezionati.

```
// Render Multiple URLs to file

var RenderUrlsToFile, arrayOfUrls, system;

system = require("system");

/*
Render given urls
@param array of URLs to render
@param callbackPerUrl Function called after finishing each URL, including the last URL
@param callbackFinal Function called after finishing everything
*/
RenderUrlsToFile = function(urls, callbackPerUrl, callbackFinal) {
    var getFilename, next, page, retrieve, urlIndex, webpage;
    urlIndex = 0;
    webpage = require("webpage");
    page = null;
    getFilename = function() {
        return "sitshow-" + urlIndex + ".png";
    };
    next = function(status, url, file) {
        page.close();
        callbackPerUrl(status, url, file);
        return retrieve();
    };
    retrieve = function() {
        var url;
        if (urls.length > 0) {
            url = urls.shift();
            urlIndex++;
            page = webpage.create();
            page.viewportSize = {
                width: 1080,
                height: 600
            };
            page.settings.userAgent = "Phantom.js bot";
            return page.open("http://" + url, function(status) {
                var file;
                file = getFilename();
                if (status === "success") {
                    return window.setTimeout((function() {
                        page.render(file);
                        return next(status, url, file);
                    })), 200);
                } else {
                    return next(status, url, file);
                }
            });
        } else {
            return callbackFinal();
        }
    };
    return retrieve();
};

arrayOfUrls = null;

// List of URLs to render as images
if (system.args.length > 1) {
    arrayOfUrls = Array.prototype.slice.call(system.args, 1);
} else {
    arrayOfUrls = [
        "roma2.rm.ingv.it",
        "ionos.rm.ingv.it/Roma/latest.html",
    ];
}
```

```

        "ionos.rm.ingv.it/gibilmanna/latest.html",
        "ionos.rm.ingv.it/tucuman/latest.html",
        "roma2.rm.ingv.it/it/risorse/osservatori_geomagnetici",
        "geomag.rm.ingv.it/index.php?op=compare", // this duplication is a
workaround to an issue saving this page
        "geomag.rm.ingv.it/index.php?op=compare",

        "roma2.rm.ingv.it/it/risorse/osservatori_geomagnetici/47/osservatorio_magnetico_di
_duronio",
        "geomag.rm.ingv.it/index.php?op=dur",

        "roma2.rm.ingv.it/it/risorse/osservatori_geomagnetici/13/osservatorio_magnetico_di
_castello_tesino",
        "geomag.rm.ingv.it/index.php?op=cts",

        "roma2.rm.ingv.it/it/risorse/osservatori_geomagnetici/25/osservatorio_magnetico_di
_stazione_mario_zucchelli",
        "geomag.rm.ingv.it/index.php?op=btd",
        "www.moist.it/sites/western_ionian_sea/2/SMO1/ctd/36/dataset/plot"
    ];
}

RenderUrlsToFile(arrayOfUrls, (function(status, url, file) {
    if (status !== "success") {
        return console.log("Unable to render '" + url + "'");
    } else {
        return console.log("Rendered '" + url + "' at '" + file + "'");
    }
}), function() {
    return phantom.exit();
});

```



# Quaderni di Geofisica

ISSN 1590-2595

<http://istituto.ingv.it/l-ingv/produzione-scientifica/quaderni-di-geofisica/>

I Quaderni di Geofisica coprono tutti i campi disciplinari sviluppati all'interno dell'INGV, dando particolare risalto alla pubblicazione di dati, misure, osservazioni e loro elaborazioni anche preliminari, che per tipologia e dettaglio necessitano di una rapida diffusione nella comunità scientifica nazionale ed internazionale. La pubblicazione on-line fornisce accesso immediato a tutti i possibili utenti. L'Editorial Board multidisciplinare garantisce i requisiti di qualità per la pubblicazione dei contributi.

# Rapporti tecnici INGV

ISSN 2039-7941

<http://istituto.ingv.it/l-ingv/produzione-scientifica/rapporti-tecnici-ingv/>

I Rapporti Tecnici INGV pubblicano contributi, sia in italiano che in inglese, di tipo tecnologico e di rilevante interesse tecnico-scientifico per gli ambiti disciplinari propri dell'INGV. La collana Rapporti Tecnici INGV pubblica esclusivamente on-line per garantire agli autori rapidità di diffusione e agli utenti accesso immediato ai dati pubblicati. L'Editorial Board multidisciplinare garantisce i requisiti di qualità per la pubblicazione dei contributi.

# Miscellanea INGV

ISSN 2039-6651

<http://istituto.ingv.it/l-ingv/produzione-scientifica/miscellanea-ingv/>

La collana Miscellanea INGV nasce con l'intento di favorire la pubblicazione di contributi scientifici riguardanti le attività svolte dall'INGV (sismologia, vulcanologia, geologia, geomagnetismo, geochimica, aeronomia e innovazione tecnologica). In particolare, la collana Miscellanea INGV raccoglie reports di progetti scientifici, proceedings di convegni, manuali, monografie di rilevante interesse, raccolte di articoli ecc..

**Coordinamento editoriale e impaginazione**

Centro Editoriale Nazionale | INGV

**Progetto grafico e redazionale**

Daniela Riposati | Laboratorio Grafica e Immagini | INGV

© 2015 INGV Istituto Nazionale di Geofisica e Vulcanologia

Via di Vigna Murata, 605

00143 Roma

Tel. +39 06518601 Fax +39 065041181

**<http://www.ingv.it>**



**Istituto Nazionale di Geofisica e Vulcanologia**