

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

**Studio e sperimentazione di strumenti
software per presentazioni pubbliche
basati su Raspberry Pi**

Relatore:
Chiar.mo Prof.
Renzo Davoli

Presentata da:
Giulia Cantini

Sessione II
Anno Accademico 2016/2017

Introduzione

Negli ultimi anni si è verificata una rapida diffusione di proiettori wireless, che sostituiscono quelli cablati, offrendo maggiore semplicità d'uso in molteplici contesti, in ambito lavorativo, per presentazioni aziendali o pubbliche, durante conferenze, eventi, o lezioni, oppure in ambiente home con lo streaming di contenuti multimediali per intrattenimento.

Uno dei difetti principali di alcuni dei modelli proposti è quello di fondarsi sull'installazione di driver proprietari e *platform-dependent* o sull'aggiunta di dispositivi hardware quali ad esempio dongle WiFi, dai quali dipende strettamente il funzionamento del proiettore.

Nel presente progetto viene realizzato un sistema di supporto alla proiezione basato sulla tecnologia VNC (Virtual Network Computing), open source e disponibile in svariate implementazioni su diverse piattaforme al fine di consentire, per quanto possibile, la stessa esperienza di utilizzo del sistema su dispositivi eterogenei.

Essenzialmente ciò che questo lavoro si propone di fare, è convertire un proiettore tradizionale in uno dotato di capacità wireless, in modo da ottenere un apparato che sia, al netto dei compromessi, comparabile in termini di efficienza e efficacia a uno dei modelli disponibili in commercio, con particolare attenzione data all'utilizzo del sistema durante presentazioni pubbliche o lezioni.

Per fare ciò, l'altra limitazione imposta, oltre alla compatibilità massima, è stata mantenere i costi di realizzazione più bassi possibile: per questa ragione l'hardware utilizzato è costituito da Raspberry Pi, in grado di aumentare le

potenzialità del proiettore con costi contenuti.

Nella seconda parte del progetto si è posta poi l'enfasi nel realizzare un meccanismo di proiezione ancora più facile ed immediato, basato sulla configurazione ad hoc di un dispositivo (Raspberry Pi Zero W) e mirato alla esclusiva proiezione di documenti PDF.

Indice

| | |
|--|-----------|
| Introduzione | 2 |
| 1 Contesto | 9 |
| 1.1 Ambiti di utilizzo | 9 |
| 1.2 VNC | 10 |
| 1.3 Raspberry Pi | 11 |
| 1.3.1 Raspberry Pi 2 | 12 |
| 1.3.2 Raspberry Pi Zero W | 12 |
| 2 Lavoro originale: Wireless Projector | 13 |
| 2.1 Panoramica e utilizzo del sistema | 13 |
| 2.2 Configurazione del server | 14 |
| 2.2.1 Windows | 14 |
| 2.2.2 macOS | 17 |
| 2.2.3 GNU/Linux | 17 |
| 2.2.4 Android/iOS | 19 |
| 2.3 Configurazione del client | 19 |
| 2.3.1 Startup del VNC client all'avvio | 19 |
| 2.3.2 Indirizzo IP statico | 21 |
| 2.3.3 Disabilitare il blank screen | 21 |
| 2.3.4 Streaming audio e video | 22 |
| 2.3.5 File system read-only | 23 |

| | | |
|----------|--|-----------|
| 3 | Lavoro originale: VNC on the GO | 31 |
| 3.1 | Introduzione | 31 |
| 3.2 | Panoramica e utilizzo del sistema | 31 |
| 3.3 | Configurazione del sistema | 32 |
| 3.4 | Piremote | 34 |
| | | |
| 4 | Valutazioni e futuri sviluppi | 37 |
| 4.1 | Estensione del supporto server per dispositivi mobili | 37 |
| 4.2 | Stabilità del canale nello streaming audio | 38 |
| 4.3 | Tempi di avvio del Raspberry Pi Zero W | 38 |
| 4.4 | Tempi di risposta in VNC on the GO | 39 |
| 4.5 | Modalità di caricamento delle presentazioni | 40 |
| 4.6 | Estensione delle possibilità di controllo tramite Piremote | 41 |
| | | |
| | Conclusioni | 41 |
| | | |
| | Bibliografia | 45 |

Elenco delle figure

| | | |
|-----|---|----|
| 1.1 | Interazione tra un client e un server VNC | 11 |
| 1.2 | Raspberry Pi 2 (a sinistra) e Zero W (a destra) | 12 |
| 2.1 | Raspberry Pi projector: panoramica del setup | 14 |
| 2.2 | TightVNC Server: Icona (rossa) del TightVNC Server nella system tray | 15 |
| 2.3 | TightVNC Server: Opzioni visualizzabili con clic destro su icon tray | 15 |
| 2.4 | TightVNC Server: Attach Listening Viewer | 16 |
| 2.5 | TightVNC Server: Screen areas | 16 |
| 2.6 | Vine Server: schermata principale | 17 |
| 2.7 | Vine Server: configurazione con reverse connection | 17 |
| 3.1 | VNC on the GO: panoramica del setup con Raspberry Pi ZeroW | 32 |
| 3.2 | Piremote: pagina principale | 35 |

Elenco delle tabelle

| | | |
|-----|--|----|
| 4.1 | Confronto tra Raspberry Pi Zero W e Raspberry Pi 3 | 40 |
|-----|--|----|

Capitolo 1

Contesto

In questo capitolo vengono illustrate le motivazioni e i presupposti sui quali si basa il progetto e vengono presentati i principali strumenti e tecnologie utilizzati nella sua realizzazione.

1.1 Ambiti di utilizzo

Analizzando la vasta scelta di proiettori wireless disponibili sul mercato diventa chiaro come non esista una tecnologia specifica su cui basare gli strumenti, anzi, in mancanza di uno standard non esiste neppure un accordo tra produttori su che cosa si intenda per "wireless". Proiettori di brand diversi vengono infatti pubblicizzati come dotati di funzionalità wireless che nella pratica utilizzano tecnologie molto distanti tra di loro, di fatto risultando in modalità di utilizzo variabili per l'utente.

Differenze in queste tecnologie riguardano molteplici fattori tra cui la tipologia di file che può essere trasmessa (documenti o video), la stabilità della connessione, il tipo di connessione richiesta, ad esempio in rete LAN privata creata ad hoc o all'interno di una rete esistente.

Solo alcuni modelli sono in grado di trasmettere video senza che l'utente noti rallentamenti nella riproduzione dell'immagine, la maggior parte infatti supporta solo la condivisione di documenti o immagini.

Quei proiettori che richiedono il setup di una rete privata per stabilire la connessione tra dispositivi, e quindi non possono collegarsi ad una rete esistente, vanno ad impegnare l'interfaccia WiFi del laptop, impedendo all'utente di avere una connessione Internet funzionante durante la proiezione, a meno che non si ricorra ad una scheda di rete esterna in aggiunta per ottenere una seconda interfaccia di rete.

La maggior parte dei modelli inoltre necessita di hardware aggiuntivo, in particolare adattatori USB, e richiede l'installazione di applicazioni sia desktop che mobile, per funzionare.

Per quanto riguarda i costi, questi variano a seconda dell'approccio seguito: è possibile avere un proiettore wireless acquistandone uno che possiede il WiFi built-in oppure si può acquistare un adattatore wireless da collegare ad un tradizionale proiettore.

Nel primo caso un proiettore wireless-enabled può costare fino a 3000\$, mentre nel secondo caso è richiesto un costo per adattatore che va dai 50\$ ai 1300\$.

La soluzione proposta di seguito si presenta come alternativa alle definizioni di "proiettore wireless" date finora, con l'obiettivo di configurarsi come una soluzione di costo minore, con ampia compatibilità di sistemi operativi e in modo particolare, volta a superare le limitazioni date da quelle tipologie di prodotti che, non integrandosi in reti esistenti, impediscono all'utilizzatore la navigazione online.

Lo strumento proposto infatti, essendo basato esclusivamente sulla tecnologia VNC per quanto riguarda la connettività, può essere collocato tra i servizi di rete che fanno uso del semplice protocollo IP per realizzare le proprie funzionalità, utilizzando un'unica interfaccia di rete fisica.

1.2 VNC

VNC (Virtual Network Computing) è una tecnologia che consente la condivisione in remoto di desktop grafici. Può trasmettere gli eventi generati

da tastiera e mouse da un ambiente grafico ad un altro, attraverso la rete, permettendo così di controllare un altro computer a distanza.

È basato sul protocollo di rete RFB (Remote Frame Buffer) che scambia i dati tra due computer ed è realizzato su un modello client/server: nell'utilizzo tipico un VNC client (o viewer) invia gli eventi da interpretare e un VNC server li riceve e li trasforma in input locali; per questo motivo, i software VNC vengono spesso utilizzati per fornire assistenza remota a pc distanti nella rete.

Uno dei punti di forza di VNC è il fatto di essere indipendente dalla piat-

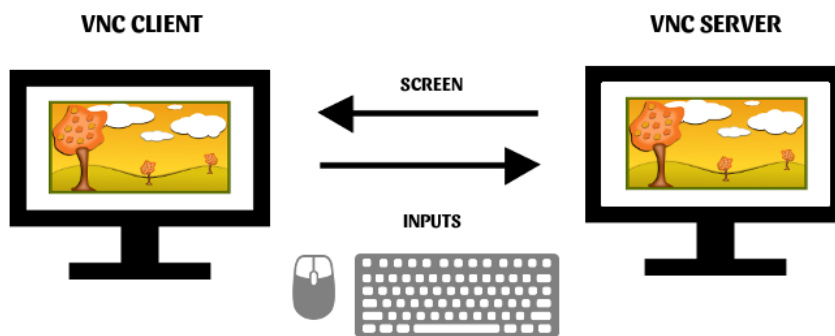


Figura 1.1: Interazione tra un client e un server VNC

taforma, esistono implementazioni software di VNC per sistemi operativi diversi, tutte capaci di interoperare tra di loro.

1.3 Raspberry Pi

I Raspberry Pi sono una serie di mini computer single-board progettati nel Regno Unito con l'intento di diffondere nelle scuole e nei paesi in via di sviluppo le basi dell'informatica e della programmazione. Dal 2012 ne sono state realizzate diverse versioni, nel progetto ne sono state utilizzate due, il Raspberry Pi 2 Model B e il Raspberry Pi Zero W.

1.3.1 Raspberry Pi 2

Fa parte della seconda generazione di Raspberry Pi, presenta una vasta dotazione di porte (USB, HDMI, Ethernet) ma manca del WiFi onboard. Nel progetto è stato dotato di WiFi attraverso un dongle USB ed utilizzato per eseguire il VNC client.

1.3.2 Raspberry Pi Zero W

È stato utilizzato nella seconda parte del progetto come VNC server. Manca di alcune porte rispetto al precedente, è di dimensioni molto minori ed è dotato di WiFi onboard.

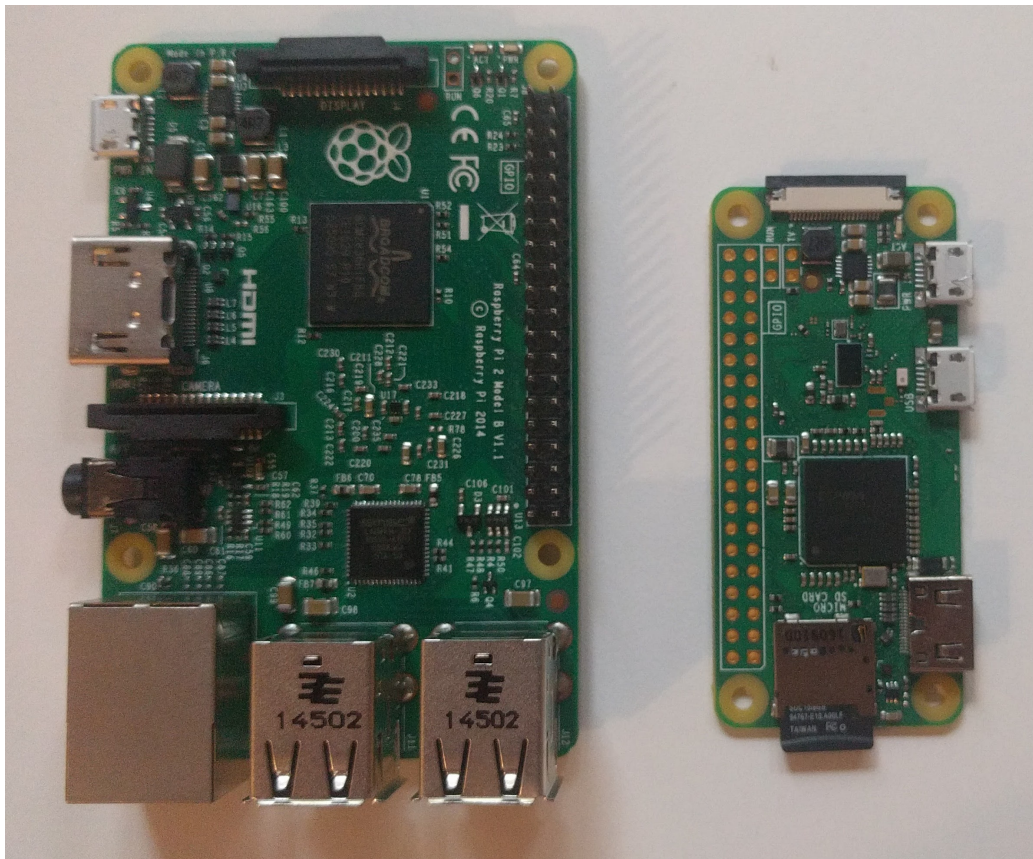


Figura 1.2: Raspberry Pi 2 (a sinistra) e Zero W (a destra)

Capitolo 2

Lavoro originale: Wireless Projector

2.1 Panoramica e utilizzo del sistema

Il sistema è formato da un client VNC in esecuzione sul Raspberry Pi 2, a sua volta collegato attraverso un cavo HDMI ad un proiettore. Si utilizza un meccanismo di connessione chiamato *reverse connection*, tipicamente è il client ad iniziare una connessione verso un server; in questo caso invece il client resta in ascolto in attesa di connessioni ed è il server VNC che viene mandato in esecuzione su un laptop all'inizio di ogni proiezione a richiedere il collegamento.

Scenari d'uso È possibile definire una descrizione di un caso d'uso tipico del sistema, da parte di un utente che sia ad esempio un relatore ad una conferenza pubblica o un professore universitario durante una lezione in aula, che utilizza comunemente il proiettore come supporto su cui mostrare delle slide o appunti che aiutino gli studenti a seguire le lezioni.

Inizialmente, il laptop personale dell'utente dovrà essere configurato seguendo un tutorial, che segue i passi descritti nelle sezioni successive, ampliandone i dettagli e che sarà reso disponibile dall'amministratore del sistema.



Figura 2.1: Raspberry Pi projector: panoramica del setup

La configurazione servirà ad installare il software necessario, e sarà obbligatoria solo al primo utilizzo, dopodiché basterà eseguire il server VNC per avere il proiettore wireless pronto all'uso.

Il Raspberry Pi 2 dovrà essere collegato al proiettore e già acceso al momento in cui il laptop apre la connessione VNC.

2.2 Configurazione del server

Il procedimento con cui si è configurato il server VNC differisce con il sistema operativo utilizzato.

2.2.1 Windows

Su sistema operativo Windows 10 è stato installato e configurato il software TightVNC rilasciato con licenza GPL2 nella versione non commerciale. Dopo l'installazione il programma è già in esecuzione e si può procedere alla configurazione: facendo clic con il tasto destro sull'icona del TightVNC Server nella system tray, apparirà una finestra popup nella quale sarà sufficiente selezionare la voce "Attach Listening Viewer".

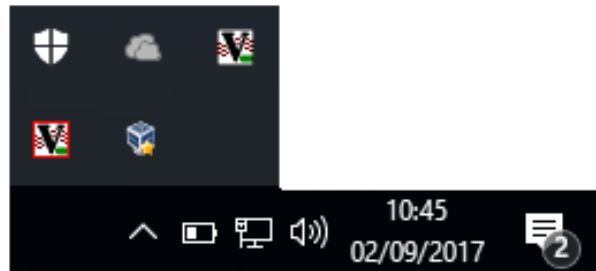


Figura 2.2: TightVNC Server: Icona (rossa) del TightVNC Server nella system tray

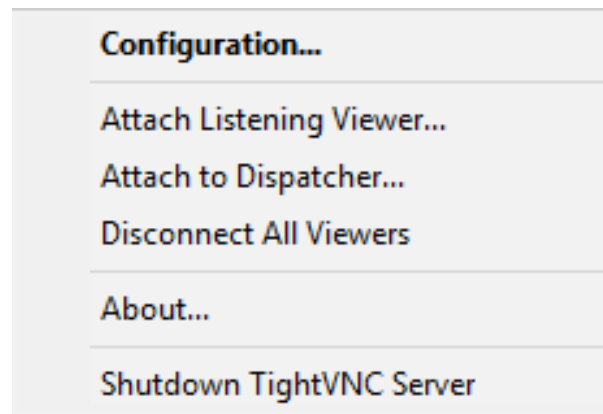


Figura 2.3: TightVNC Server: Opzioni visualizzabili con clic destro su icon tray

Comparirà un'ulteriore finestra, nella quale sarà possibile specificare un indirizzo e una porta TCP per un client in ascolto.

Fare tick su "Restricted view-only access" per evitare che il client possa agire modificando il desktop del server. Se viene rilevato un client in listening all'indirizzo specificato, viene avviata immediatamente la condivisione dello schermo.

È stato inoltre aggiunto un hostname nel file *C:\Windows\system32\drivers\etc\hosts* per identificare il Raspberry Pi 2 all'interno della sottorete.

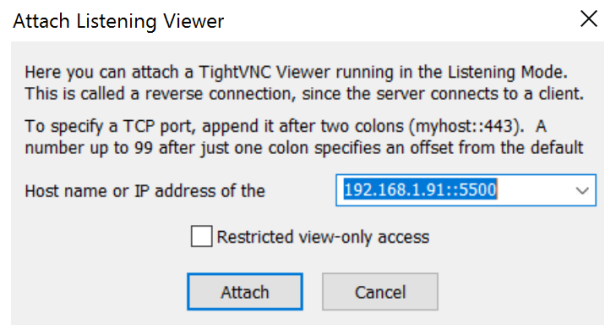


Figura 2.4: TightVNC Server: Attach Listening Viewer

Con TightVNC è possibile limitare la porzione di schermo da virtualizzare utilizzando una delle opzioni del pannello di configurazione, o in alternativa, da linea di comando usando

```
1 tvnserver.exe -sharerect widthxheight+left+top
```

considerando che l'origine (0,0) del piano è posizionata all'angolo in alto a sinistra del display primario.

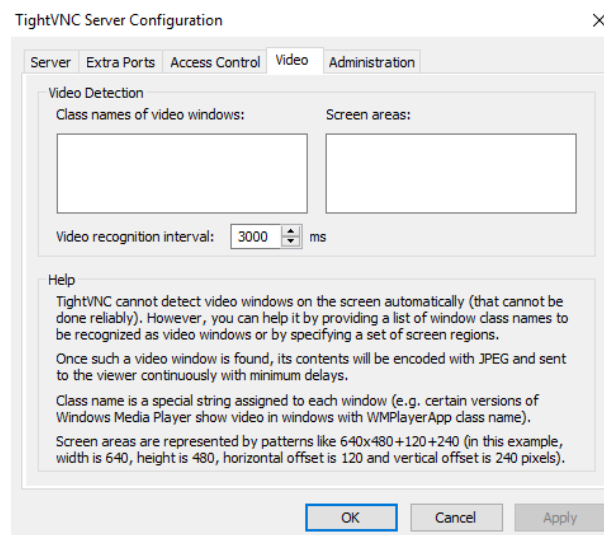


Figura 2.5: TightVNC Server: Screen areas

2.2.2 macOS

Su sistema operativo macOS Sierra si è utilizzato il software Vine Server (OSXvnc), che è open source e distribuito con licenza GPL3.

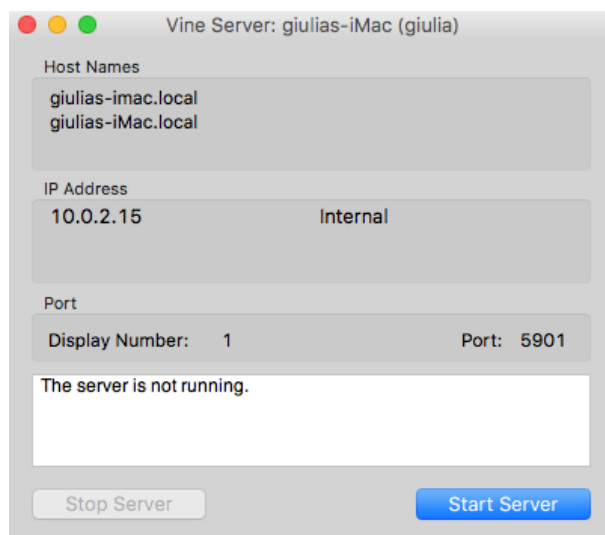


Figura 2.6: Vine Server: schermata principale

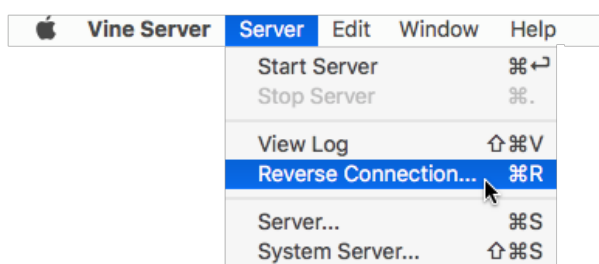


Figura 2.7: Vine Server: configurazione con reverse connection

2.2.3 GNU/Linux

Il testing è avvenuto su distribuzione Ubuntu alla versione 16.04 Xenial Xerus.

x11vnc È un server VNC ricco di funzionalità e possibilità di personalizzazione, facilmente acquisibile perché presente come package nei repository ufficiali:

```
1 sudo apt-get install x11vnc
```

Una connessione viene istanziata dal server x11vnc lanciando semplicemente il comando

```
1 x11vnc -connect <client-ip-address>:<port>
```

con cui il server si collega al client in listening mode specificato da indirizzo e porta TCP.

Con x11vnc è inoltre possibile modificare la risoluzione del pannello VNC per adattarla a quella del proiettore in modo da ottenere la visione migliore. È possibile infatti usare l'opzione *-geometry WxH* oppure l'opzione *-scale factor* per specificare nel primo caso larghezza ed altezza in pixel, e nel secondo un fattore con cui scalare il framebuffer.

```
1 x11vnc -geometry 1600x1200
2 x11vnc -scale 0.8
```

Un'altra opzione utile è *-clip WxH+X+Y*, che consente di ridurre l'area di schermo da visualizzare sul client. Questa opzione può essere combinata con le precedenti per riadattare la risoluzione dello schermo una volta applicato il ridimensionamento del pannello.

```
1 x11vnc -clip 800x600+0+0 -geometry 1920x1080
```

Anche in questo caso è possibile aggiungere un hostname che identifichi il Raspberry Pi 2 all'interno della sottorete, è sufficiente modificare (con privilegi da amministratore) il file */etc/hosts* aggiungendo la linea

```
1 192.168.1.78      raspberrypi.localdomain raspberrypi
```

avendo sostituito all'indirizzo quello corrente del client.

2.2.4 Android/iOS

Sebbene fosse inizialmente tra gli obiettivi del progetto quello di includere tra i sistemi operativi supportati anche quelli mobili - data la crescente diffusione di questi ultimi anche nel supporto di presentazioni pubbliche - date le molte difficoltà incontrate, non si sono trovate soluzioni soddisfacenti.

In Android sono presenti o soluzioni parziali come server che non supportano reverse connection, oppure soluzioni che richiedono il rooting del dispositivo da parte dell'utente.

In iOS l'unico server disponibile (Veency) è utilizzabile solo da chi ha effettuato il jailbreak del proprio dispositivo, e d'altra parte Apple non sembra interessata a supportare VNC anche in mobile, dove per la condivisione in remoto di contenuti è già disponibile la suite AirPlay.

2.3 Configurazione del client

Il Raspberry Pi 2 è dotato di sistema operativo Raspbian, basato su distribuzione Debian nella sua versione per architetture ARM.

La configurazione del client può essere suddivisa in sottosezioni.

2.3.1 Startup del VNC client all'avvio

Considerando che il Raspberry Pi 2 viene utilizzato come dispositivo headless - senza tastiera e mouse collegati - è necessario che dopo il setup iniziale, ogni programma venga avviato in modo automatico. Per abilitare il client all'avvio è stato utilizzato un metodo che si avvale di desktop entry e della

presenza di X in corretta esecuzione.

È un metodo supportato dai maggiori *desktop environment*, tra i quali LXDE e il suo derivato PIXEL, disponibile in Raspbian.

È necessario innanzitutto abilitare il boot con il desktop

```
1 sudo raspi-config
```

Selezionando

Enable boot to desktop/Scratch >

Desktop Log in as user 'pi' in the graphical desktop .

Dopodiché creiamo un file .desktop (o desktop entry) nella directory usata dall'ambiente desktop per l'autostart delle applicazioni; quella di default, a livello utente è `/.config/autostart/`:

```
1 cd /home/pi/.config
2 mkdir autostart
3 cd autostart
4 nano tightvnc.desktop
```

Scrivendo nel file di configurazione:

```
1 [Desktop Entry]
2 Type=Application
3 Name=TightVNC
4 Exec=vncviewer -viewonly -fullscreen -listen
5 StartupNotify=false
```

Tramite il campo Exec viene specificato il comando da eseguire all'avvio. In questo modo il VNC viewer sarà messo in ascolto ed eseguito senza che possa modificare il desktop del server e in modalità a schermo intero. Infine è necessario salvare e riavviare per apportare le modifiche

```
1 sudo reboot
```

2.3.2 Indirizzo IP statico

La configurazione di un indirizzo IP (privato) statico sul client è utile per fare in modo che i server conoscano in anticipo l'indirizzo del client, prevedendo che comunque il Raspberry Pi sia utilizzato sempre nella stessa sottorete.

È necessario modificare `/etc/dhcpd.conf`, il file di configurazione letto dal demone di *dhcpd*, un client DHCP e DHCPv6 open source.

```
1 interface wlan0
2 static ip_address=192.168.1.78/24
3 static routers=192.168.1.254
4 static domain_name_servers=192.168.1.254
```

2.3.3 Disabilitare il blank screen

Questo passo è necessario per evitare che il desktop grafico del Raspberry vada in standby interrompendo la presentazione.

È sufficiente modificare il file di configurazione del display manager (LightDM) `/etc/lightdm/lightdm.conf` aggiungendo nella sezione `[SeatDefaults]` il comando

```
1 [SeatDefaults]
2 xserver --command=X -s 0 -dpms
```

2.3.4 Streaming audio e video

PulseAudio È un server audio che funge da proxy per tutte quelle applicazioni che utilizzano l'audio tramite componenti come ALSA e OSS. Una delle caratteristiche più interessanti di PulseAudio è che può essere utilizzato via rete per fare streaming audio da un client ad un server sul quale è in esecuzione il demone PulseAudio, permettendo la riproduzione del suono sul server.

Per prima cosa è necessario abilitare lo streaming attraverso TCP per client anonimi.

Nel file di configurazione `/etc/pulse/default.pa` abilitare i seguenti parametri sul server (Pi):

```
1 load-module module-native-protocol-tcp auth-ip-acl  
   =127.0.0.1;192.168.0.0/24 auth-anonymous=1
```

Dove la sottorete LAN (192.168.0.0/24) deve coincidere con quella in cui si trovano i client che richiedono di connettersi.

Sul client (laptop), aggiungere la riga

```
1 load-module module-native-protocol-tcp
```

Successivamente, per fare in modo che il server PulseAudio compaia nel PulseAudio Device Chooser (pasystray), si devono caricare i moduli zeroconf necessari e abilitare il demone Avahi.

Sia sul server che sul client

```
1 sudo apt install pulseaudio-module-zeroconf  
2 sudo systemctl start avahi-daemon.service  
3 sudo systemctl enable avahi-daemon.service
```


Sul server, in `/etc/pulse/default.pa` è necessario aggiungere la riga

```
1 load-module module-zeroconf-publish
```

Mentre sul client, in `/etc/pulse/default.pa`

```
1 load-module module-zeroconf-discover
```

Ora per ridirezionare un qualsiasi stream o output audio al server PulseAudio remoto è sufficiente selezionare il sink appropriato ad esempio utilizzando PulseAudio Volume Control (pavucontrol).

Se non appaiono i sinks sul client, provare a riavviare il demone Avahi sul server per rifare il broadcast delle interfacce disponibili.

```
1 sudo systemctl restart avahi-daemon.service
```

2.3.5 File system read-only

Prevedendo un utilizzo continuativo del Raspberry Pi, e non potendo assicurarsi sempre un corretto spegnimento del dispositivo, è stato necessario trovare un metodo che potesse limitare al minimo i rischi di danneggiare la micro SD sulla quale il sistema è installato.

Una delle soluzioni è quella di minimizzare le scritture su memoria SD, che potrebbero lasciare il file system in uno stato incoerente ad esempio in caso di repentini cali di tensione o spegnimenti erranei.

A tale scopo, si è predisposto un meccanismo che rende possibile l'utilizzo in read-only delle partizioni sensibili (di sistema) e garantisce comunque la possibilità di mantenere e amministrare il sistema stesso perché consente uno switch in read-write ogni volta che l'amministratore debba installare nuovi programmi o apportare modifiche nei file di configurazione.

UnionFS È un file system Linux che permette *union mount*, ovvero consente di simulare l'unione di più file system sottostanti. File e directory in file system separati, denominati *branch* vengono sovrapposti in modo trasparente, determinando un unico file system coerente.

I diversi branch possono essere sia in sola lettura che in scrittura e lettura: in questo caso le scritture nella copia unificata vengono indirizzate verso uno specifico e reale file system. Questo permette ad un file system di apparire come scrivibile anche se in realtà non lo è.

Un esempio di utilizzo di questo particolare meccanismo (*copy-on-write*) si ha nei Live CD.

Nel progetto è stato utilizzato unionfs-fuse, una implementazione userspace di UnionFS, che presenta la stesse caratteristiche.

È stato utilizzato per sovrapporre ad un sistema in sola lettura, un file system temporaneo memorizzato in RAM; ciò comporta che ogni scrittura effettuata avvenga in RAM e non nella memoria SD, con conseguente perdita di ogni informazione aggiunta ad ogni riavvio del Raspberry.

In questo modo si previene che eventuali scritture possano danneggiare o modificare il file system in modo inappropriato.

unionfs-fuse è disponibile come pacchetto nei repository di Raspbian:

```
1 apt-get update
2 apt-get install unionfs-fuse
```

È necessario disabilitare tutti i servizi che possono scrivere in modo incontrollato su micro SD, tra i quali anche il servizio che si occupa della memoria swap:

```
1 dphys-swapfile swapoff
2 dphys-swapfile uninstall
3 update-rc.d dphys-swapfile disable
4
5 systemctl disable systemd-readahead-collect
6 systemctl disable systemd-random-seed
```

Inoltre creiamo uno script per potere utilizzare UnionFS:

```
1 touch /usr/local/bin/mount_unionfs
2 chmod +x /usr/local/bin/mount_unionfs
3 nano /usr/local/bin/mount_unionfs
```

In cui scriveremo

```
1 #!/bin/sh
2 DIR=$1
3 # check if root is rw or ro
4 ROOTMOUNT=$(awk ' $2==" /" {print substr($4,1,2)}' < /etc/fstab)
5 if [ $ROOTMOUNT = "rw" ]
6 then # remount the first directory in the second
7     /bin/mount --bind ${DIR}_org ${DIR}
8 else
9     # remount the directory as tmpfs or ramdisk
10    /bin/mount -t tmpfs ramdisk ${DIR}_rw
11    /usr/bin/unionfs-fuse -o cow,allow_other,suid,dev,nonempty ${
12    DIR}_rw=${DIR}_org=${DIR}
13 fi
```

Ora modifichiamo le partizioni di sistema nella tabella dei file system

```
1 nano /etc/fstab
```

aggiungendo le opzioni che imposteranno il readonly:

```
1 proc /proc proc defaults 0 0
2 /dev/mmcblk0p1 /boot vfat ro 0 2
3 /dev/mmcblk0p2 / ext4 ro,noatime 0 1
4 mount_unionfs /etc fuse defaults 0 0
5 mount_unionfs /var fuse defaults 0 0
```

```
6 none /tmp tmpfs defaults 0 0
```

Dopodiché è necessario modificare il file `cmdline.txt` per montare `/boot` in `readonly`:

```
1 nano /boot/cmdline.txt
```

Si aggiunge 'ro' dopo il `root=/dev..` come in esempio:

```
1 dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty1 root=/
  dev/mmcblk0p2 ro rootfstype=ext4 elevator=deadline rootwait
  fbcon=map:1 fbcon=font:ProFont6x11
```

Creiamo uno script per uscire dal `readonly` nel caso dovessimo installare del software e mantenere la modifica anche dopo un riavvio:

```
1 nano /usr/local/bin/noreadonly
```

```
1 #!/bin/bash
2
3 # remount root rw
4 mount -o remount,rw /
5
6 # prepare target paths
7 mkdir -p /chroot
8 mkdir -p /chroot/{bin,boot,dev,etc,home,lib,opt,proc,root,run,
  sbin,sys,tmp,usr,var}
9
10 # mount special file systems
11 mount -t proc proc /chroot/proc
12 mount --rbind /sys /chroot/sys
13 mount --rbind /dev /chroot/dev
14
15 # bind rw directories
```

```
16 for f in {etc,var}; do mount --rbind /${f}_org /chroot/$f; done
17
18 # bind remaining directories
19 for f in {bin,boot,home,lib,opt,root,run,sbin,tmp,usr}; do mount
    --rbind /$f /chroot/$f; done
20
21 # chroot
22 echo "Note: /boot is still mounted read-only, remount to read-
    write if needed."
23 echo -e "\e[33mYou are now in read-write chroot. Use CTRL+D when
    done to exit chroot and mount read-only again.\e[39m"
24 # enter chroot in the shell
25 chroot /chroot /usr/bin/env PS1="(rw) \u@\h:\w\$ " /bin/bash --
    noprofile -l
26
27 # unmount mounts
28 for f in /chroot/{bin,boot,dev,etc,home,lib,opt,proc,root,run,
    sbin,sys,tmp,usr,var}; do
29 umount -l $f
30 done
31
32 sleep 1
33
34 # remount read-only again
35 echo -e "\e[32mChroot left, re-mounting read-only again.\e[39m"
36 mount -o remount,ro /
```

Lo script essenzialmente effettua il remounting della partizione / in lettura-scrittura: inoltre, dato che il package manager dovrà scrivere anche nelle directory /etc e /var, non è sufficiente fare remount di /, per questa ragione nello script si utilizza chroot per cambiare root directory e successivamente si usano dei bind mount per popolare l'albero chroot/.

Chroot È un'operazione che modifica in apparenza la directory root per il processo in esecuzione ed i suoi figli. Un programma eseguito in tale ambiente non può accedere a file e comandi al di fuori dell'albero delle directory di quell'ambiente, che viene denominato *chroot jail*. Per questa caratteristica, non c'è il rischio che il processo vada a scrivere in parti del filesystem che sono attualmente montate in read-only.

Bind mount È un'operazione che crea una vista alternativa per un determinato albero delle directory. Differisce da un mount classico nel fatto che mentre questo costruisce un nuovo albero a partire da un certo device, un bind mount prende un albero esistente e lo replica in un altro punto del file system.

L'albero originale e la replica conterranno gli stessi dati e ogni modifica ad uno si ripercuoterà sull'altro.

Diamo allo script i permessi di esecuzione

```
1 chmod +x /usr/local/bin/noreadonly
```

È inoltre necessario preparare le nuove directory, creando dei symbolic link per quei programmi che devono scrivere su file system, come dhcp:

```
1 rm -rf /var/lib/dhcp/  
2 ln -s /tmp /var/lib/dhcp  
3 rm /etc/resolv.conf  
4 ln -s /tmp/resolv.conf /etc/resolv.conf  
5 cp -al /etc /etc_org  
6 mv /var /var_org  
7 mkdir /etc_rw  
8 mkdir /var /var_rw
```

Inoltre poiché X11 non funziona correttamente con la /home in readonly, è necessario mettere in atto il seguente workaround:

```
1 mkdir /var_org/home
2 mv /home/pi /var_org/home/pi
3 chmod -R 777 /var_org/home/pi
4 cd /home
5 ln -s /var/home/pi
6 reboot
```

Dopo il riavvio il sistema apparirà in read-only, tutti i file che verranno creati adesso spariranno al riavvio successivo.

Nel caso siano richieste invece delle modifiche permanenti basterà lanciare lo script noreadonly appena creato e all'interno della chroot jail invocare i comandi necessari.

Capitolo 3

Lavoro originale: VNC on the GO

3.1 Introduzione

Nella seconda parte viene descritta l'estensione del progetto iniziale a comprendere un dispositivo di supporto alla proiezione che funzioni out-of-the-box; un Raspberry Pi Zero W facilmente trasportabile e con tempi di boot ridotti.

In un tipico utilizzo basterà avviare il mini computer collegandolo ad un power bank, e dopo la sequenza di avvio, il PDF per la presentazione sarà già pronto per essere discusso, mentre il controllo delle slide avverrà tramite smartphone collegato ad un'applicazione web.

3.2 Panoramica e utilizzo del sistema

Descriviamo il funzionamento del sistema distinguendo tra due utilizzi principali: un utilizzo wireless, come vera e propria estensione della prima parte del progetto, il Wireless Projector, e un utilizzo wired.

In entrambi i setup il Raspberry Pi Zero W è headless e alimentato da un po-

wer bank con capacità di 10.000 mAh, più che sufficiente a coprire l'intervallo temporale occupato da una classica presentazione o lezione.

Uso wireless In questo setup, il laptop è sostituito dal Raspberry Pi Zero W su cui è installato e configurato x11vnc così come descritto nella sezione "Configurazione del server - GNU/Linux".

Dall'altra parte è necessario che sia presente e correttamente configurato un Raspberry Pi 2 Wireless Projector che accetti la richiesta di connessione VNC in entrata.

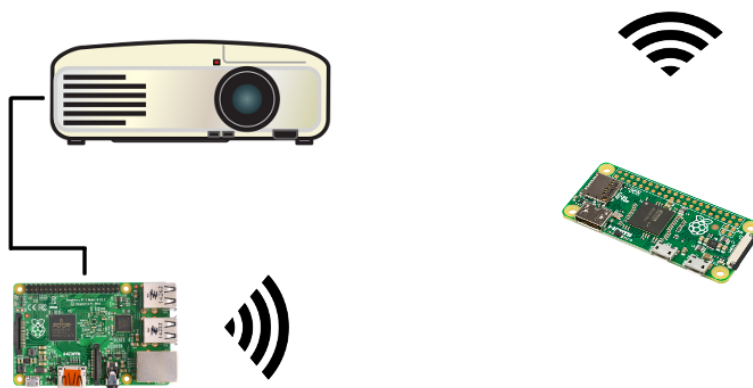


Figura 3.1: VNC on the GO: panoramica del setup con Raspberry Pi ZeroW

Uso wired Il Raspberry Pi Zero W è collegato al proiettore via cavo HDMI o VGA tramite adattatore, secondo la disponibilità di porte del proiettore stesso. Questo consente di utilizzare le funzionalità sviluppate nell'estensione anche nei casi in cui si disponga unicamente di proiettori tradizionali.

3.3 Configurazione del sistema

È stata predisposta una cartella `/home/pi/Presentation` destinata a contenere i documenti PDF da visualizzare durante una sessione di proiezione,

la presentazione di standard aperta all'avvio è denominata "default.pdf", da questa successivamente si potranno aprire le altre eventuali presentazioni presenti nella directory.

Si prevede che il caricamento dei file su Raspberry Pi Zero W possa avvenire in vari modi, ad esempio tramite chiavetta USB, lettore di SD card o protocollo SCP.

Sul sistema si trova uno script (/home/pi/startup.sh) che chiamato all'avvio tramite .desktop file ha tre compiti: avviare la connessione VNC (per uso wireless), eseguire l'applicazione web Piremote, eseguire il programma Evin- ce con la presentazione base.

Piremote, che viene eseguita dallo script, è contenuta di default in /home/pi.

Di seguito sono riportati i dettagli della configurazione:

```
1 cd /home/pi/.config
2 mkdir autostart
3 cd autostart
4 nano startup.desktop
```

Scrivere nel file di configurazione (apportando eventuali modifiche):

```
1 [Desktop Entry]
2 Type=Application
3 Name=Startup
4 Exec=lxterminal -e /home/pi/startup.sh
5 Terminal=true
```

```
1 sudo nano /etc/hosts
2 >> 192.168.1.73      projector.localdomain      projector
```

Il contenuto dello script startup.sh:

```
1 #! /bin/bash
2
3 x11vnc --connect projector:5500 &
4 sleep 10s
5 cd /home/pi/piremote
6 python -m piremote.main &
7 sleep 10s
8 evince /home/pi/Presentation/default.pdf &
```

La configurazione per uso *wired* è molto simile; l'unica differenza è che non sarà necessario eseguire anche VNC all'avvio.

3.4 Piremote

Come visualizzatore di documenti PDF è stato scelto Evince, un programma semplice e leggero disponibile nella distribuzione. Per controllare la presentazione si è realizzato un semplice controller accessibile attraverso un browser web da un qualsiasi dispositivo mobile (o desktop) collegato alla stessa LAN in cui si trova il Pi Zero W. Con tale applicazione è possibile trasformare il proprio smartphone ad esempio, in un telecomando del tutto simile ad un presenter.

L'applicazione è molto semplice e supporta le operazioni di base per il controllo della presentazione: lo scorrimento delle pagine avanti e indietro, l'apertura di un file .pdf a partire da una directory designata (/home/pi/Presentation) e il passaggio a schermo intero nella modalità di presentazione.

Per poter accedere a tutte le funzionalità è necessario inserire nella schermata iniziale un pin o una password per autenticarsi, dopodiché si verrà reindirizzati alla pagina principale.

Piremote è stata rilasciata con licenza MIT ed è disponibile all'indirizzo <https://github.com/giulic3/piremote>.

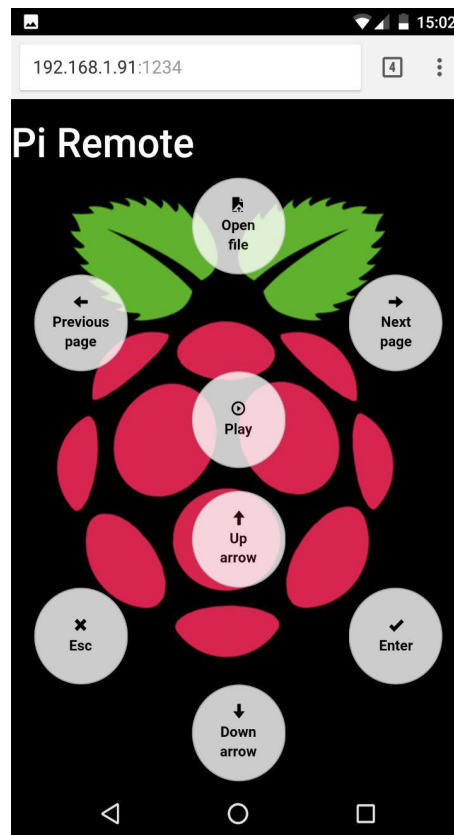


Figura 3.2: Piremote: pagina principale

Dall'alto verso il basso: il bottone "Open file" consente di aprire un file PDF che si trova all'interno della stessa directory in cui si trova la presentazione corrente;

"Previous page" e "Next page" permettono di scorrere le pagine della presentazione corrente;

premendo "Play" è possibile passare alla modalità presentazione, a schermo intero;

le frecce "Up arrow" e "Down arrow" consentono la navigazione della directory all'apertura di un file;

con "Esc" è possibile uscire dalla modalità presentazione;

con "Enter" si seleziona un file PDF.

Capitolo 4

Valutazioni e futuri sviluppi

Di seguito viene riportata un'analisi del contributo originale, suddivisa in un'analisi generale del progetto inteso come somma delle sue due parti e un'analisi specifica indirizzata in modo particolare al lavoro su VNC on the GO.

4.1 Estensione del supporto server per dispositivi mobili

Come riferito nella sottosezione "2.2.4 Android/iOS", uno degli aspetti migliorabili del presente progetto riguarda il supporto per dispositivi mobili, per i quali attualmente sono assenti implementazioni come software libero di server VNC.

Il supporto sarebbe infatti caratteristica gradita in quanto, con la rapida crescita del mercato mobile a discapito di quello desktop, non è più raro trovare relatori che usano i propri tablet per proiettare le loro presentazioni, e in questi casi il sistema proposto non può più essere applicato.

Allo stato delle cose comunque, trovare una soluzione a questo problema sembra molto difficile, a meno che non si abbandoni l'uso di VNC per passare ad altri protocolli.

Nel mondo Android, le poche implementazioni VNC disponibili richiedono

il rooting del dispositivo, che non è una procedura universale e consigliabile per ogni utente, per varie ragioni, tra cui sicurezza, rischio di rovinare il dispositivo, invalidazione della garanzia.

Nel mondo iOS poi, difficilmente un'app come un VNC server, che potenzialmente trasmette in chiaro dati sensibili in remoto, potrebbe mai essere approvata per essere pubblicata nello store ufficiale.

4.2 Stabilità del canale nello streaming audio

Durante i test svolti per controllare il funzionamento dello streaming audio/video da server a client si è notato una differenza della stabilità del flusso audio in riproduzione, variabile insieme al software che funge da sorgente audio.

Si sono effettuati test che hanno coinvolto il browser Mozilla Firefox e i lettori musicali VLC, Amarok, Banshee, Rhythmbox. I risultati migliori si sono ottenuti con quest'ultimo.

Inoltre, la riproduzione audio è stata talvolta soggetta a interruzioni e variazioni di volume, probabilmente imputabili a una scorretta comunicazione tra client e server. Questi problemi devono essere investigati ulteriormente provando diverse configurazioni di PulseAudio su più macchine server.

4.3 Tempi di avvio del Raspberry Pi Zero W

Un aspetto fondamentale per avere una esperienza di utilizzo positiva del sistema, è garantire che i tempi di boot siano i più rapidi possibile, per poter iniziare la presentazione in tempi ragionevoli. Per questo motivo sul mini-computer è stato installato Raspbian in versione Lite, priva di software aggiuntivi che normalmente sono inseriti nella versione completa, e privo di interfaccia grafica.

Solo successivamente si sono installati i pacchetti (minimi) necessari per abilitare il desktop grafico, richiesto per l'utilizzo di VNC.

Sono stati inoltre disabilitati alcuni service in background ritenuti inutili per l'utilizzo previsto.

Per misurare i tempi di boot si è utilizzato il comando `systemd-analyze` che viene comunemente usato per misurare statistiche sulla performance di un sistema all'avvio, il quale ha riportato in output che lo startup termina in 1.498 secondi in kernel space e in 23.841 secondi in user space, per un totale di 25.340 secondi.

I risultati ottenuti sono sembrati accettabili, tuttavia, è possibile fare di meglio: per velocizzare ulteriormente l'avvio, si potrebbe utilizzare un sistema operativo con kernel custom dotato strettamente dei moduli necessari, utilizzando ad esempio un tool come *Buildroot* per la compilazione.

4.4 Tempi di risposta in VNC on the GO

Durante i test svolti per valutare l'usabilità del sistema VNC on the GO dal punto di vista dell'esperienza utente, uno degli aspetti principali è stato analizzare la reattività del sistema agli input inviati tramite Piremote durante una normale presentazione. Inizialmente, l'applicazione è stata testata mentre veniva eseguita su laptop Dell Inspiron 7537. Il controllo della presentazione avviene con risposte immediate, con un ritardo non percepibile dall'utente.

Successivamente, si è testato il sistema con Piremote in esecuzione su Raspberry Pi Zero W, con e senza l'utilizzo di VNC.

Nel primo caso, si nota immediatamente un'aggiunta di ritardo nella reazione agli input da web browser: principalmente dovuta alle limitate risorse del mini-computer rispetto al laptop.

L'esecuzione del server VNC in aggiunta, introduce un ulteriore overhead, dato dal fatto che ad ogni cambio di pagina o apertura di file, è necessario che la schermata lato server sia completamente ridisegnata sul client.

Un possibile miglioramento potrebbe prevedere la sostituzione del Raspberry Pi Zero W, che è molto più efficiente se utilizzato in contesti che non prevedano l'utilizzo di ambiente grafico, con un modello di Raspberry con dotazione superiore (la versione 3 ad esempio), accettando il compromesso di dover aumentare il costo complessivo del progetto.

| | Raspberry Pi Zero W | Raspberry Pi 3 (Model B) |
|------------------|----------------------------|--------------------------------------|
| CPU | BCM2835 1GHz ARM11 | Quadcore 64bit 1.2GHz ARM Cortex A53 |
| GPU | none | Broadcom VideoCore IV @ 400 MHz |
| RAM | 512 MB | 1GB (LPDDR2-900 SDRAM) |
| Storage | MicroSD | MicroSD |
| USB | 1 Micro USB socket | 4 |
| Ethernet | 0 | 1 |
| Wi-Fi | 802.11n Wireless LAN | 802.11n Wireless LAN |
| Bluetooth | Bluetooth 4.0 | Bluetooth 4.0 |
| Video out | Mini-HDMI | HDMI/Composite via RCA Jack |
| Audio out | none | 3.5mm jack |
| GPIO | 40 (unpopulated) | 40 |
| Size | 65 mm x 30 mm x 5 mm | 85.6 mm x 56.5 mm x 17 mm |
| Price | \$10 | \$35 |

Tabella 4.1: Confronto tra Raspberry Pi Zero W e Raspberry Pi 3

4.5 Modalità di caricamento delle presentazioni

All'interno della sezione precedente si sono descritte le modalità con cui una presentazione viene scelta per l'esecuzione a partire da una directory prefissata. Tra le possibilità contemplate per il trasferimento di tale presentazione sul Raspberry Pi Zero W si è parlato di collegamento tramite USB, utilizzando lettore di SD card oppure per copia tramite protocollo SCP. In un'estensione del progetto sarebbe interessante aggiungere la possibilità

per l'utente di ottenere le presentazioni in automatico all'avvio direttamente dalla rete, facendo uso ad esempio di un servizio di archiviazione cloud quale Dropbox, Mega o Google Drive.

Il sistema dovrebbe essere sincronizzato con un account specifico e al momento dell'avvio dovrebbe essere configurato per effettuare il download della presentazione desiderata, trasferita sul cloud precedentemente.

Tale meccanismo richiede una connessione di rete attiva, che potrebbe essere resa disponibile dal Raspberry stesso in caso sia possibile collegarsi a rete libera oppure si abbia modo di autenticarsi in una rete non libera, altrimenti si potrebbe semplicemente utilizzare lo smartphone come hotspot a cui associare il Raspberry via tethering, e quindi poter usufruire della rete cellulare, in sostituzione al WiFi.

4.6 Estensione delle possibilità di controllo tramite Piremote

Allo stato attuale Piremote implementa solamente le funzionalità di base coinvolte nel controllo di una presentazione, per cui esistono molteplici altre operazioni per le quali sarebbe possibile aggiungere il supporto. Ad esempio, le modalità con il quale avviene il caricamento della prima presentazione e la ricerca delle altre è attualmente limitato: la ricerca dei documenti PDF può avvenire unicamente all'interno della directory designata, mentre sarebbe più conveniente un metodo che consentisse di navigare liberamente il file system per trovare la presentazione desiderata.

Conclusioni

In questo progetto si è realizzato un sistema di supporto alla proiezione di contenuti multimediali come slide, o tracce audio e video, durante presentazioni pubbliche o lezioni. Il sistema si è basato su VNC e sull'utilizzo di Raspberry Pi e nella prima parte si è concentrato sulla configurazione di un laptop personale come VNC server, mentre nella seconda è stata proposta un'estensione che realizzasse l'interazione client-server out-of-the-box. Si è poi effettuata un'analisi del funzionamento del progetto e dei risultati attesi, che si sono rilevati positivi ma ampiamente suscettibili di miglioramenti.

Bibliografia

- [1] Larry L. Peterson, Bruce S. Davie. "Computer Networks: A Systems Approach" . Morgan Kaufmann, 2011.
- [2] Abraham Silberschatz, Peter B. Galvin, Greg Gagne. "Operating System Concepts". John Wiley & Sons, Inc., 2013.
- [3] "Wireless Projector Guide". <https://airtame.com/wireless-projector>. Ultima visita: 15 settembre 2017.
- [4] "x11vnc: a VNC server for real X displays". http://www.karlrunde.com/x11vnc/x11vnc_opts.html. Ultima visita: 15 settembre 2017 .
- [5] "Raspberry Pi Documentation". [https://www.raspberrypi.org/ documentation/](https://www.raspberrypi.org/documentation/). Ultima visita: 15 settembre 2017.
- [6] "Debian Wiki". <https://wiki.debian.org/it/FrontPage>. Ultima visita: 15 settembre 2017.
- [7] "Desktop entries". https://wiki.archlinux.org/index.php/desktop_entries. Ultima visita: 15 settembre 2017.
- [8] "PulseAudio/Examples". [https://wiki.archlinux.org/index.php/ PulseAudio/Examples_\(Italiano\)](https://wiki.archlinux.org/index.php/PulseAudio/Examples_(Italiano)). Ultima visita: 15 settembre 2017.
- [9] "RaspberryPI in Readonly". <https://guglio.xyz/raspberrypi-in-readonly/>. Ultima visita: 15 settembre 2017.

- [10] "Make Raspbian System Read-Only".
<http://blog.pi3g.com/2014/04/make-raspbian-system-read-only/>.
Ultima visita: 15 settembre 2017.
- [11] "Tornado". <http://www.tornadoweb.org/en/stable/> . Ultima visita: 15 settembre 2017.
- [12] "Improve boot performance". [https://wiki.archlinux.org/index.php/Improve_boot_performance_\(Italiano\)](https://wiki.archlinux.org/index.php/Improve_boot_performance_(Italiano)). Ultima visita: 15 settembre 2017.

Ringraziamenti

Ringrazio la mia famiglia che mi ha sostenuto moralmente durante questo faticoso percorso durato tre anni.

Ringrazio Lorenzo che mi sopporta ogni giorno e condivide con me le gioie e i dolori della vita universitaria.

Ringrazio gli amici e i compagni di corso che mi hanno generosamente aiutato nelle diverse occasioni.

Infine ringrazio il professore Renzo Davoli per il tempo dedicatomi e i gentili consigli.