



SAET I.S. s.r.l
Via Leinì, 1/B
10077 S. Maurizio Canavese (TO)
Tel. +39 0119375208
Fax. +39 0119278846

SPECIFICA TECNICA

Nome

**Architettura S.O. VarcoLAN
SV**

AUTORE	DESCRIZIONE	DATA	REV.
Luca Mini	Creazione	18/04/2016	A

	Nome	Data	Firma
Revisionata			
Approvata			

Razionale

Descrizione dell' architettura del File System e gestione dell'applicazione all'interno della memoria SD del Supervisore, basato su scheda Raspberry PI 2

Indice

1 - Legenda, unità di misura usate e nomenclatura.....	3
1.1 - Legenda e nomenclatura.....	3
1.2 - Unità di misura usate.....	3
2 - Creazione dell'immagine.....	3
2.1 - Preparazione SD di partenza.....	3
2.2 - installazione delle applicazioni di sistema.....	3
2.3 - abilitare la seriale.....	3
2.4 - real time clock.....	4
2.5 - creare FS read only e aggiunta area dati.....	4
2.6 - abilitazione /tmp.....	5
2.7 - creazione delle cartelle di lavoro in /data.....	6
2.8 - creare link simbolici a file di sistema.....	6
2.9 - configurazione apache.....	7
2.10 - Regola di UDEV per lettore di badge USB.....	7
2.11 - gestione dell'esecuzione automatica.....	8
2.12 - modifica indirizzo ip statico.....	8
2.13 - abilitazione WiFi (opzionale).....	8
2.14 - Velocizzare SSH.....	9
2.15 - abilitazione del Watchdog.....	10
2.16 - Configurazione PHP.....	10
2.17 - Configurazione firewall.....	10
2.18 - Creazione immagine finale.....	11
3 - Architettura.....	11
3.1 - Modalità di lavoro.....	11
3.1.a - Start-up applicazione.....	11
3.1.b - backup dei dati su SD.....	12
3.2 - Modalità aggiornamento.....	12
3.2.a - creare un pacchetto di aggiornamento.....	12
3.2.b - scaricare un nuovo aggiornamento del supervisore.....	12
4 - Riferimenti.....	12

1 - Legenda, unità di misura usate e nomenclatura

1.1 - Legenda e nomenclatura

NA indica che la richiesta non è applicabile

TBD “to be defined”, soggetto ancora da definire

1.2 - Unità di misura usate

Salvo ove non specificatamente indicato si utilizza il Sistema Internazionale (SI)

2 - Creazione dell'immagine

Si parte dal sistema operativo originale *2016-02-26-raspbian-jessie-lite.img* a cui vengono rimosse/installate applicazioni ed apportate modifiche per renderlo, a parte alcune aree, Read Only in modo da evitare scritture dannose per la scheda SD,

2.1 - Preparazione SD di partenza

Eseguire i passi descritti nella guida *SD_prepare.pdf*.

Disinserire e reinserire la SD, ed eseguire Gparted

Si dovrebbero vedere 2 partizioni *mmcblk0p1* e *mmcblk0p2*, e lo spazio non usato della SD.

Selezionare lo spazio non usato e creare una nuova partizione, poi formattarla *ext4*. La nuova partizione avrà nome *mmcblk0p3*.

2.2 - installazione delle applicazioni di sistema

```
$ sudo apt-get install apache2 php5 libapache2-mod-php5 sqlite php5-sqlite python-smbus i2c-tools  
raspi-config
```

2.3 - abilitare la seriale

La seriale si chiama */dev/ttyAMA0* (in RPI3 *ttyS0*)

occorre disabilitare il servizio

```
sudo systemctl stop serial-getty@ttyAMA0.service
```

```
sudo systemctl disable serial-getty@ttyAMA0.service
```

questo invece per cambiare gruppo, in modo da poter essere gestito da utente normale:

```
sudo chgrp dialout /dev/ttyAMA0
```

```
sudo chmod 660 /dev/ttyAMA0
```

(probabilmente questo va messo in *init.d*)

2.4 - real time clock

Mettere il programma rtc-spi in /usr/bin

Per rimettere l'ora:

```
# rtc_spi YYYYMMGGhhmmss
```

per impostarla nel sistema

```
# rtc_spi
```

NOTA: questo programma per sincronizzare l'ora viene eseguito nel servizio in init.d (vedi più avanti)

disabilitare l'NTP server

```
sudo update-rc.d -f ntp remove
```

Abilitare la risincronizzazione dell'orologio mediante processo cron

scrivere il file *varco_time* in */etc/cron.daily* (oppure in */etc/cron.hourly*) con scritto:

Creare un crontab per rimettere l'ora ogni giorno come da RTC

```
$ sudo crontab -e
```

selezionare l'editor e aggiungere la riga:

```
0 12 * * * /usr/bin/rtc_spi
```

ogni giorno alle 12.00 l'orologio di sistema sarà aggiornato come l'rtc hw

2.5 - creare FS read only e aggiunta area dati

All'inizio è possibile accedere mediante ssh al sistema con:

```
# ssh pi@<ipaddress>
```

password: *raspberry*

che successivamente sarà cambiata in “*saetis*”

da qui eseguire quanto riportato nella guida *SD_safe.pdf*.

Eseguire i seguenti ulteriori comandi

```
# mkdir /data
```

```
# chmod 777 /data
```

```
# vim /etc/fstab
```

modificare il file così:

```
proc          /proc          proc    defaults    0    0
```

```

/dev/mmcblk0p1 /boot      vfat  defaults,ro      0    2
/dev/mmcblk0p2 /          ext4  defaults,noatime,ro 0    1
/dev/mmcblk0p3 /data      ext4  defaults,noatime    0    2
# a swapfile is not a swap partition, no line here
# use dphys-swapfile swap[on|off] for that
tmpfs /var/log      tmpfs  nodev,nosuid 0    0
tmpfs /var/tmp       tmpfs  nodev,nosuid 0    0
tmpfs /tmp          tmpfs  nodev,nosuid 0    0

# sync

```

2.6 - abilitazione /tmp

Sembra che la cartella tmp per come viene montata dal sistema non abbia i i diritti per consentire all'utente di scriverci, per cui occorre creare uno script che smonta e rimonta la partizione

Creare un file /etc/init.d/varco.sh con scritto

```

umount /tmp

mount /tmp

ln -s /tmp /var/log/apache2

# to set RTC

rtc_spi

```

modificare il file /etc/init.d/apache2 aggiungendo in fondo alla riga:

```
# Required-Start: ..... $varco.sh
```

eseguire il comando:

```

# update-rc.d varco.sh defaults
# update-rc.d apache2 defaults

```

in caso di problemi con le sessioni, occorre modificare *php.ini* in */etc/php5/apache2/php.ini*

```
session.save_path = "/tmp/sessions"
```

(lo script di lancio del SV deve creare la cartella sessions in tmp)

2.7 - creazione delle cartelle di lavoro in /data

Assicurarsi di non essere root (usare exit se il prompt è #)

\$ cd /data

CARTELLA	PERMESSI	DESCRIZIONE
database	drwxrwxrwx	Contiene tutti i database attuali
database_bkp	drwxrwxr-x	Backup dei database quando sono pieni o per aggiornamenti
firmware	drwxrwxr-x	Tutti i firmware caricati
firmware_new	drwxrwxr-x	Cartella che contiene i firmware nuovi che forzano gli aggiornamenti delle periferiche. Normalmente è vuota, viene riempita con i pacchetti che verranno trasferiti in firmware
firmware_sched	drwxrwxrwx	Contiene gli aggiornamenti schedulati non ancora attuati. Normalmente è vuota, viene riempita con i pacchetti che verranno trasferiti in firmware
sv_scripts	drwxrwxr-x	Contiene script per il supervisore
sys	drwxrwxr-x	Mantiene file di sistema che vengono cambiati dal programma. Sono per lo più linkati in cartelle read only
update	drwxrwxr-x	Aggiornamenti del supervisore
utils	drwxrwxr-x	Contiene script di utilità
webapp	drwxrwxr-x	Web application
.backup	drwxr-xr-x	backup .tar.gz delle versioni precedenti
.backup/temp		Creata temporaneamente per l'installazione dallo script install.sh

In /data ci saranno i seguenti file

varcosv eseguibile

varcosv.cfg configurazione dell'applicazione

sv_startup.sh script di esecuzione e respawn (\$ chmod 755 sv_startup.sh)

2.8 - creare link simbolici a file di sistema

Quando il FS sarà read-only non sarà più possibile modificare niente che riguarda il sistema per cui occorre portare in area modificabile tutti quei file che invece dovranno essere modificati, tra cui il file che configura l'indirizzo IP del sistema

```
# mv /etc/network/interfaces /data/sys
```

```
# ln -s /data/sys/interfaces /etc/network/interfaces
```

```
# sync
```

Attenzione: il file viene modificato dall'applicazione con “sed” per cui occorre stare attenti al suo contenuto di partenza, ovvero, una sola interfaccia definita, ad esempio:

```
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd

# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'
```

```
# Include files from /etc/network/interfaces.d:
```

```
source-directory /etc/network/interfaces.d
```

```
auto lo
```

```
iface lo inet loopback
```

```
#iface eth0 inet manual
```

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.10.100
```

```
netmask 255.255.255.0
```

```
network 192.168.10.0
```

```
broadcast 192.168.10.255
```

```
gateway 192.168.10.1
```

```
dns-nameservers 8.8.8.8 8.8.4.4
```

2.9 - configurazione apache

Dentro la cartella /var/www creare un link simbolico alla cartella /data/webapp/

```
$ sudo ln -s /data/webapp/ /var/www/html
```

aggiungere l'utente pi al gruppo di apache:

```
$ sudo usermod -a -G www-data pi
```

2.10 - Regola di UDEV per lettore di badge USB

Per vedere i dati salienti dell'interfaccia connessa, collegare il lettore e dare il comando:

```
$ udevadm info /dev/ttyUSBx
```

Aggiungere un file alle regole di udev:

```
# vim /etc/udev/rules.d/99_usb_serial.rules
```

con il seguente contenuto

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001",  
ATTRS{product}=="USB <-> Serial", SYMLINK+="ttyReader"
```

il dispositivo quando verrà connesso, sarà accessibile tramite /dev/ttyReader

2.11 - gestione dell'esecuzione automatica

Aggiungere al file */etc/rc.local* le seguenti righe

```
rtc_spi
```

```
chmod 666 /dev/watchdog
```

```
/data/sv_startup.sh &
```

e copiare lo script *sv_startup.sh* in */data*

2.12 - modifica indirizzo ip statico

```
# vim /etc/network/interfaces
```

e aggiungere al posto della riga “iface eth0 inet manual ”

```
#iface eth0 inet manual
```

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.10.69
```

```
netmask 255.255.255.0
```

```
network 192.168.10.0
```

```
broadcast 192.168.10.255
```

```
gateway 192.168.10.1
```

2.13 - abilitazione WiFi (opzionale)

Individuare la rete wifi a cui collegarsi

```
sudo iwlist wlan0 scan
```


per eseguire una scansione delle reti, da cui rilevare l'ESSID e il tipo di cifratura usata, ad esempio
IE: IEEE 802.11i/WPA2 Version 1

na volta individuato l'essid e la cifratura modificare il file

```
# vim /etc/wpa_supplicant/wpa_supplicant.conf
```

e al fondo aggiungere

```
network={
    ssid="The_ESSID"
    psk="Your_wifi_password"
}
```

ovviamente coi dati dela rete da attivare.

Editare il file `/etc/network/interfaces` per assegnare un IP statico (se necessario):

e aggiungere (modificare) le seguenti linee, adeguandole alla rete:

```
allow-hotplug wlan0
```

```
#iface wlan0 inet manual
```

```
iface wlan0 inet static
```

```
address 192.168.30.44
```

```
netmask 255.255.255.0
```

```
network 192.168.30.0
```

```
broadcast 192.168.30.255
```

```
gateway 192.168.30.1
```

```
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

2.14 - Velocizzare SSH

Modificare il file di configurazione

```
# vim /etc/ssh/sshd_config
```

aggiungere alla fine del file:

```
UseDNS no
```

2.15 - abilitazione del Watchdog

Aggiungere al file /etc/modules la riga

```
bcm2708_wdog
```

per far caricare il modulo ad ogni avvio.

Installare il demone

```
# apt-get install watchdog
```

eseguire il comando per farlo avviare ad ogni avvio:

```
# update-rc.d watchdog defaults
```

oppure

```
# chkconfig --add watchdog
```

aprire il file

```
vim /etc/watchdog.conf
```

decommentare la riga che inizia per #watchdog-device per poter usare il dispositivo.

Decommentare la riga che dice #max-load-1 = 24 per riavviare se il carico va oltre la quota stabilita per oltre 1 minuto. NOTA: un carico 25 per un minuto significa che servirebbero 25 raspberry pi per completare il lavoro in 1 minuto.

Avviare il servizio:

```
# chkconfig watchdog on
```

oppure

```
# /etc/init.d/watchdog start
```

2.16 - Configurazione PHP

Per permettere l'upload dei pacchetti di aggiornamento occorre rimuovere il limite a 2M

```
vim /etc/php5/apache2/php.ini
```

e modificare il valore a

```
upload_max_filesize = 4M
```

2.17 - Configurazione firewall

TODO

/application_bkp preventivamente ripulito, poi passa all'installazione dei nuovi files

3. copia il database varcolan_events in /tmp (che risiede in ram) [eseguito dall'applicazione]
4. esegue l'applicazione

3.1.b - backup dei dati su SD

Onde evitare che la SD si possa danneggiare, i dati frequentemente modificabili risiedono in /tmp, e vengono salvati dall'applicazione stessa, su SD card a tempi prestabiliti.

3.2 - Modalità aggiornamento

In questa modalità è possibile caricare la nuova applicazione

3.2.a - creare un pacchetto di aggiornamento

- Mettere in 11_web i file dell'applicazione web
- compilare il programma applicativo varcosv (progetto rasp_varcosv) in configurazione Debug
- aggiornare gli script necessari in 08_scripts, in particolare il file install.sh
- eseguire lo script createpkg.sh

verrà creato un pacchetto del formato *varcosv_MMmm.tar.gz* (con MM = major version; mm = minor version) nella cartella *01_software/packages*. Se è presente un pacchetto precedente verrà salvato in *01_software/packages/old_packages*

3.2.b - scaricare un nuovo aggiornamento del supervisore

I nuovi aggiornamenti vanno messi in ~/update

I file in questa cartella vengono esaminati ad ogni avvio della macchina. Il formato del pacchetto avrà un nome fisso *varcolansv_MMmm.tar* dove MMmm sono la versione (major, minor).

Il file deve contenere almeno i seguenti files:

varcolansv	(applicazione)
varcolan.txt	(MD5 e versione del nuovo software)
install.sh	(script installer che verrà eseguito dallo script di startup)

ovviamente potrà contenere anche altri files (database, configurazioni, ecc)

4 - Riferimenti

1. Creazione preliminare della SD [SD_prepare.pdf]
2. Creazione della SD con FS read-only [SD_safe.pdf]

- | | |
|-------------------------------------|--|
| 3. Descrizione sintetica Varco Lan: | [Features_summary.pdf] |
| 4. Specifica VarcoLAN | [tech_spec_varcolan.pdf] |
| 5. Diagrammi di flusso Varcolan: | [Sync_profiles_weektime_flowchart.pdf] |