VENIAMO AL SOFTWARE

Nello scorso fascicolo abbiamo realizzato il primo script CGI con cui abbiamo testato il funzionamento del web server di Web-Sentinel. Passiamo ora agli script di azionamento del robot.

Come hai potuto vedere nel Workshop, precedente **è possibile inviare comandi alla scheda di controllo dei servocomandi usando semplicemente le istruzioni basilari della shell di Linux.** Ora vediamo come è possibile realizzare i principali script di controllo di Web-Sentinel sfruttando proprio le potenzialità dell'interfaccia testuale di Linux. **Tutti gli script descritti in queste pagine dovranno essere salvati nella directory della Fox Board e resi eseguibili con l'uso del comando 'chmod', come mostrato a pagina 5 del fascicolo precedente.**

Per il funzionamento del robot è necessario creare una serie di script e alcuni file di memorizzazione delle variabili di stato del robot.

PRIMA DI COMINCIARE ›››

Prima di passare alla scrittura degli script veri e propri è indispensabile **creare tre file di stato** (vedi immagine sotto) che permetteranno al robot di **tenere memoria dello stato del sistema e dei parametri operativi.** Come lo script di test creato in precedenza, anche questi file dovranno essere contenuti nella cartella **'/etc/httpd/cgi'** della Fox Board. Vediamoli più nel dettaglio. Il primo file deve essere creato con il nome **'seriale'** e **deve contenere al suo interno il nome della porta seriale** che intendiamo utilizzare per far comunicare la Fox Board con la MiniServoBoard. Nel nostro caso tale nome sarà, quindi, la stringa **'/dev/ttyS3'** (qui, come negli altri casi, senza apici). Gli altri due file, invece, sono chiamati **'posh'** e **'posv'** e memorizzano i **valori di posizionamento dei motori della microcamera.** Nel caso ideale, **in fase di inizializzazione questi file conterranno il valore 128, in modo da avviare il robot con la microcamera centrata,** ma tale valore potrà essere variato a piacimento per correggere le imprecisioni di montaggio delle squadrette della testa.

LO SCRIPT DI INIZIALIZZAZIONE ›››

Il primo script che analizziamo è quello che dovrà essere eseguito all'avvio del robot, in quanto **contiene la procedura di inizializzazione dell'hardware del robot.** Tale script è mostrato e commentato nel box in alto a destra della pagina successiva (eccetto la prima riga, necessaria a indicare l'interprete dello script, **le righe che iniziano con il simbolo**

192.168.0.92 - PuTTY

```
[root@axis-00408cae02e4 /etc/httpd/cgi]722# ls
avanza          posh            retrocedi       turncamdown     turnleft
dato.txt        posv            seriale         turncamleft     turnright
initrobot       prova.html      stop            turncamright
killall         prova.sh        test            turncamup
[root@axis-00408cae02e4 /etc/httpd/cgi]722#
```

'#' sono da intendersi come commenti, come da sintassi Linux). Come detto nel paragrafo precedente, questo script deve essere creato con il nome 'initrobot' e salvato nella cartella '/etc/httpd/cgi', rendendolo poi eseguibile con il comando chmod, come mostrato.

LO SCRIPT DI STOP ›››
Lo script di stop ha la funzione di bloccare i servocomandi e di 'uccidere' i processi inerenti al movimento di Web-Sentinel. Lo script deve essere archiviato assieme al precedente, nominato 'stop' e reso eseguibile. Nel box in basso a destra puoi vedere il contenuto di questo script.

LO SCRIPT DI MOVIMENTO ›››
Veniamo ora agli script di movimento del robot. Innanzitutto incominciamo dicendo che avremo a che fare con otto script differenti, quattro dei quali dedicati al movimento del robot su strada e altri quattro all'azionamento dei motori della telecamera. Poiché lo spazio a disposizione è particolarmente ridotto, verranno mostrati solo alcuni di questi script come esempi di riferimento: gli altri potranno essere facilmente ricostruiti a partire da questi. Iniziamo con lo script di avanzamento, chiamato 'WS_avanza', mostrato in alto nella pagina successiva (tutti gli script di movimento dovranno aver nome che inizia con il prefisso 'WS_', in modo da poter essere terminati con il comando 'killall WS_*' visto all'interno dello script 'stop'). Gli altri script di movimento su strada saranno praticamente identici: l'unica differenza sarà relativa alle linee evidenziate nel box, che dovranno essere modificate in funzione del comportamento desiderato. È così, allora, che per la retromarcia (script 'WS_retrocedi') dovremo

lo script 'initrobot'

```
#! /bin/sh
#avvio della webcam (come mostrato nel fascicolo 85 a pag. 8 e 9)
spcaload
servfox -d /dev/video0 -s 640x480 -w 7070

#rimozione dei tre file di stato
rm posh
rm posv
rm seriale

#Impostazione dei valori di default per l'inizializzazione del #sistema
#Questi valori vanno modificati per centrare la webcam
def_posh=128
def_posv=128
def_seriale=/dev/ttyS3

#ricreazione dei tre file
touch posh
touch posv
touch seriale

#scrittura del contenuto delle variabili di inizializzazione
#del robot delle variabili di stato
echo $def_posh > posh
echo $def_posv > posv
echo $def_seriale > seriale

#inizializzazione della porta seriale
#(come mostrato nel WS90)
stty -F $def_seriale 9600 -echo
```

lo script 'stop'

```
#! /bin/sh

#'uccidiamo' tutti i processi relativi al movimento del robot
killall WS_*

#ricarichiamo il nome della porta seriale e inviamo ai motori 2 e 3
#della MiniServoBoard il comando di stop
seriale=`tail seriale`
echo "\ro2\r">$seriale
echo "\ro3\r">$seriale
```

lo script 'WS_avanza'

```
#! /bin/sh

#carichiamo nella variabile 'seriale' il contenuto del file 'seriale'
#NB: è necessario usare gli apici 'dritti', non gli apostrofi.
seriale=`tail seriale`

#inviamo alla MiniServoBoard i comandi di direzionamento dei
#servocomandi di movimento del robot (entrambi avanzano, ruotando secondo
#versi opposti in quanto disposti in maniera simmetrica)
echo "\rM2020\r">$seriale
echo "\rM3200\r">$seriale


#inviamo alla MiniServoBoard il comando di azionamento
#dei servocomandi 2 e 3 utilizzando il comando 02 e 03.
echo "\r02\r">$seriale
echo "\r03\r">$seriale

#'uccidiamo' lo script WS_avanza (si autotermina) tramite il comando 'kill'
#che riceve il PID del processo ottenuto con il comando pidof
kill $(pidof WS_avanza)
```

far indietreggiare entrambi i servocomandi (echo "\rM2200\r">$seriale e echo "\rM3020\r">$seriale) e dovremo cambiare l'ultima riga in 'kill $(pidof WS_retrocedi)'. In ugual modo, la rotazione a destra ('WS_ruotadestra') deve attivare il motore sinistro in direzione di avanzamento tenendo spento il destro, mentre la rotazione verso sinistra ('WS_ruotasinistra') deve essere realizzata in modo duale al precedente.

IL MOVIMENTO DELLA TELECAMERA

Passiamo ora ad analizzare gli script di controllo del movimento della telecamera. Innanzitutto è indispensabile ricordare un particolare di fondamentale importanza per la programmazione: mentre i due servo delle ruote sono a rotazione continua e per il loro controllo è sufficiente 'sbilanciare' l'impulso verso uno dei due estremi di movimento, i due motori di controllo della telecamera devono essere controllati in posizione in modo corretto. Durante i movimenti è indispensabile tenere conto della posizione attualmente assunta dal motore. Normalmente questo compito sarebbe svolto da una variabile. Il problema sorge, però, dal fatto che la shell Linux è sì in grado di gestire l'uso delle variabili, ma la memorizzazione dei dati è limitata al tempo di vita del processo 'sh', che viene terminato all'esecuzione di ogni script. Utilizzeremo, quindi, le due variabili 'posh' e 'posv' citate in precedenza. Vediamo come esempio lo script 'WS_camerasu', che alza la telecamera (pagina successiva). In questo esempio vengono utilizzate anche alcune strutture di controllo (if/fi) del linguaggio della shell. Per maggiori informazioni sulla sintassi e sui comandi per la programmazione della shell Linux ti rimandiamo alla documentazione facilmente reperibile su Internet. Il file duale deve essere chiamato 'WS_cameragiu' e deve operare in modo analogo, ma opposto (sempre agendo sulla posizione del motore 0 tramite una serie di incrementi del valore di 'posv' e controllando che non raggiunga un finecorsa opposto a quello attualmente utilizzato per lo script precedente). Un discorso identico riguarda il movimento laterale della web cam. I due script, chiamati rispettivamente 'WS_cameradestra' e 'WS_camerasinistra', sono strutturalmente identici a quello appena osservato. L'unica differenza riguarda, ovviamente, il motore azionato, che sarà il numero 1. Durante la loro

scrittura è di conseguenza necessario variare tutte le righe in cui vengono generati i comandi per la MiniServoBoard. Ad esempio, la riga 'echo "M0$posv\r">>$seriale' dovrà diventare 'echo "M1$posv\r">>$seriale'. Per sperimentare il funzionamento degli script appena creati, è sufficiente che ti colleghi alla Fox Board via Telnet e ti posizioni nella cartella '/etc/httpd/cgi'. Successivamente puoi eseguire i singoli script digitando il comando './nomecomando' (ovviamente privo di apici).

lo script 'WS_camerasu'

```
#! /bin/sh

#carichiamo il contenuto del file 'posv' nella variabile omonima
posv=`tail posv`

#carichiamo il contenuto del file 'seriale'
#nella variabile di memorizzazione
seriale=`tail seriale`

#se 'posv' è maggiore di 15 allora non si è ancora arrivati al valore di
# fine corsa e il movimento può essere applicato
if [ $posv -gt 15 ]
then
#applichiamo il movimento decrementando l'attuale valore di 'posv' di 10 unità
  posv=$(($posv - 10))
fi

#inviamo il comando alla MiniServoBoard in base all'attuale valore della #posizione del
#motore
#Se il valore attuale della posizione è maggiore di 99
#allora è formato da 3 cifre
if [ $posv -gt 99 ]
then
#inviamo il comando di movimento al motore 0.
#il comando inizia con i caratteri M0 (movimento del motore 0) e viene
#seguito dalle tre cifre del valore di posizionamento
   echo "M0$posv\r">>$seriale
else
#se il valore della posizione è maggiore di 9 e minore di 100
#allora è composto da 2 cifre.
   if [ $posv -gt 9 ]
   then
#inviamo il comando di movimento al motore 0.
#il comando inizia con i caratteri M0 (movimento del motore 0) e viene
#seguito da uno 0 e dalle due cifre del valore di posizionamento
      echo "M00$posv\r">>$seriale
   else

#in alternativa, viene generato il comando per la posizione minore di 10
#(ossia a singola cifra). Il comando è generato partendo da M0 a cui seguono
#due zeri e la cifra associata alla posizione
      echo "M000$posv\r">>$seriale
   fi
fi

#rimuoviamo il file posv
rm posv

#ricreiamo il file posv
touch posv

#copiamo nel file posv il valore della variabile posv
echo $posv > posv
```