

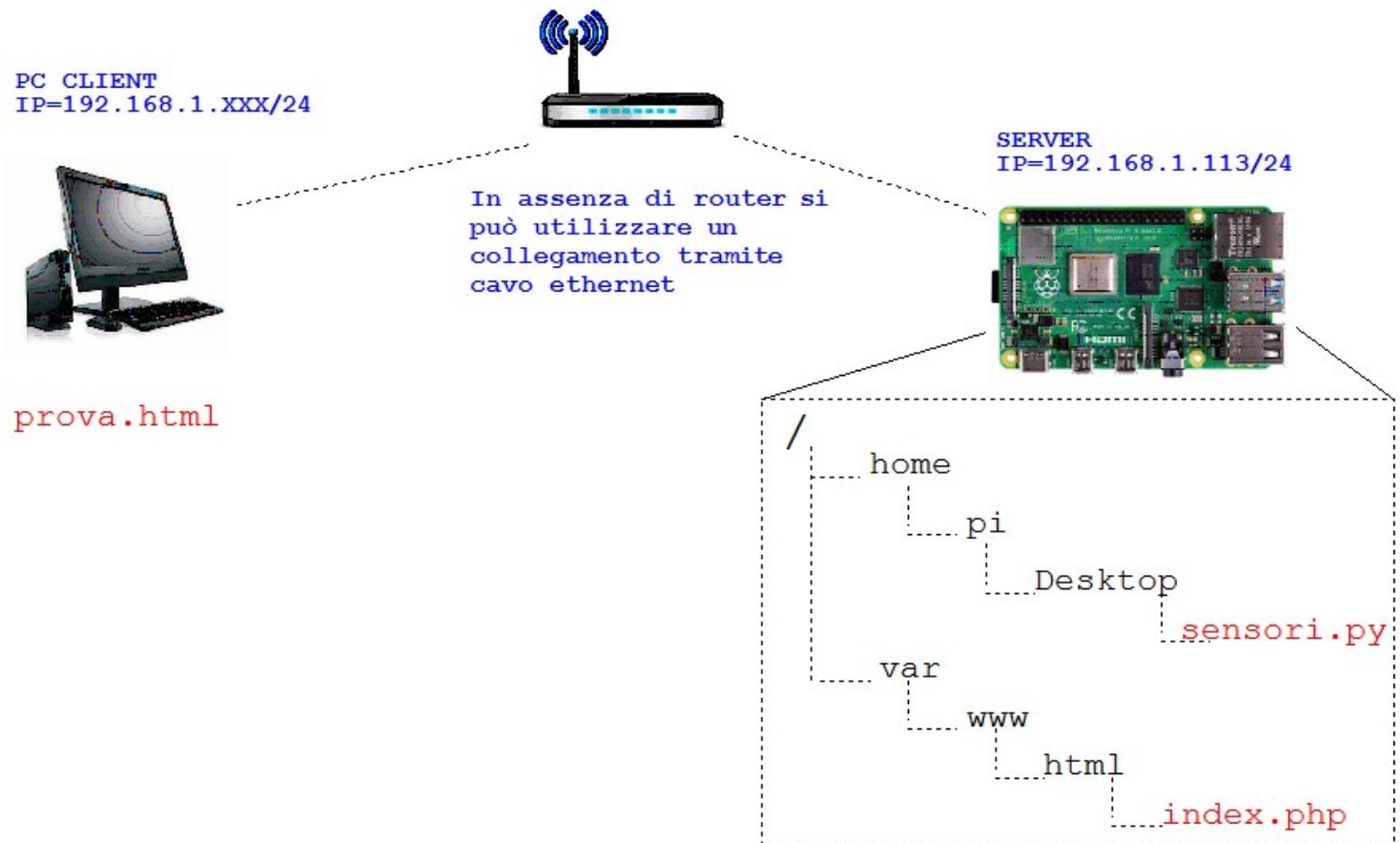
ESEMPIO DI WEBSERVER CON RASPBERRY

Per realizzare il seguente esercizio, dobbiamo avere una scheda Raspberry con installato LAMP (vedi dispensa “**Webserver con Raspberry**”) ed una scheda Sense Hat (vedi dispensa “**Raspberry Sense Hat**”).

L'idea è quella di accedere alla scheda Raspberry da remoto, per ricevere informazioni sul valore di 3 sensori: Temperatura, Pressione ed Umidità, decidendo inoltre il colore della scritta che visualizzerà il valore della grandezza scritta sul display a matrice di led presente sulla scheda Sense Hat.

Occorre innanzitutto, realizzare un sistema come quello di figura, dove il PC Client ed il Server Raspberry, risultano collegati insieme tramite cavo Ethernet o tramite una rete wifi.

Nel nostro caso l'indirizzo della scheda è: 192.168.1.113, tale valore ovviamente sarà differente e dovrà essere inserito nel file prova.html presente nel PC client.



Nel server Raspberry dovremo invece copiare il file index.php nel percorso **/ var / www / html**

ed il file sensori.py nel percorso **/ home / pi / Desktop**

Si consiglia per questa operazione di accedere alla scheda Raspberry, tramite putty o tramite realVnc viewer, come indicato nella dispensa “**Raspberry Introduzione**”.

I 2 file possono essere copiati sul desktop tramite una normale penna USB o tramite il client FTP Filezilla.

Una volta copiati sul desktop, si dovrà copiare il file index.php nel percorso indicato lanciando la CLI e digitando quanto segue:

```
cd Desktop
```

```
sudo cp index.php /var/www/html
```

Vediamo ora il contenuto dei 3 file.

1) File `prova.html` presente nel PC client

```
<html>
<body>
<form action="http://192.168.1.113/index.php" method="get">
<pre>
P=Pressione T=Temperatura U=Umidità
SENSORE:   <input type="text" name="sensore"><br>

COLORI LED RGB VALORE MAX 255
ROSSO:     <input type="text" name="rosso"><br>
VERDE:     <input type="text" name="verde"><br>
BLU:       <input type="text" name="blu"><br></pre>

<input type="submit" value="INVIA">
</form>

</body>
</html>
```

Il codice sopra riportato, genera la pagina a fianco, dove risultano presente un form con dei campi da compilare associati alle **4 variabili** indicate nel file: **sensore, rosso, verde, blu**.

Il file dovrà essere aperto con un qualsiasi browser.

Con il pulsante **INVIA**, le 4 variabili vengono passate al file **index.php** presente nel server che si trova all'indirizzo indicato cioè 192.168.1.113. Questo indirizzo ovviamente va modificato in base al proprio caso.

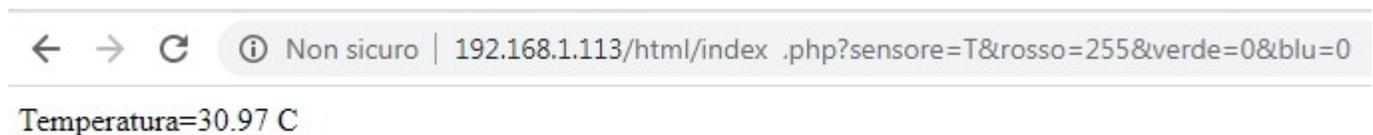
P=Pressione T=Temperatura U=Umidità
SENSORE:

COLORI LED RGB VALORE MAX 255
ROSSO:

VERDE:

BLU:

L'invio avviene con il metodo **GET**, che renderà visibili le variabili ed il loro contenuto sulla barra dei comandi. Dopo aver premuto il tasto invia il file `index.php` verrà eseguito e sulla barra degli indirizzi del nostro browser appariranno le variabili ed il loro contenuto, mentre sullo schermo troveremo le risposte del file `php`.



2) File `index.php` presente nel Server Raspberry al percorso `/var/www/html`

```
<?php
$val1 = $_GET['sensore'];
$val2 = $_GET['rosso'];
$val3 = $_GET['verde'];
$val4 = $_GET['blu'];
$command="sudo python /home/pi/Desktop/sensori.py $val1 $val2 $val3 $val4";
exec($command, $out, $status);
echo $out[0];
echo $out[1];
echo $out[2];
?>
```

Tramite il metodo **GET** le variabili vengono passate a questo file, che le copia in ordine nelle sue variabili interne: **val1**, **val2**, **val3** e **val4**.

L'istruzione `$_GET` serve infatti a leggere il contenuto della variabile inviata. Nel linguaggio PHP le variabili sono seguite dal carattere `$`.

Successivamente all'interno del file, viene generata la variabile `$command`. Questa variabile conterrà la stringa necessaria per lanciare il file `sensori.py` da CLI e cioè:

***sudo python /home/pi/Desktop/sensori.py** con di seguito le 4 variabili*

Dal file php, si potrà lanciare programma `sensori.py` usando il comando `exec`, che non fa altro che eseguire la stringa creata da terminale, come fossimo noi a scriverla manualmente.

Se il file lanciato, scriverà a video delle stringhe, queste verranno ricevute nella variabile `out` che rappresenta in realtà un array di valori.

Nel nostro caso, aspettiamo dal file 3 `print`, il primo rappresenta la scritta della grandezza richiesta, il secondo il suo valore ed il terzo la sua unità di misura.

es. *Temperatura=*
 30,97
 C

Le 3 stringhe verranno restituite al client con 3 semplici `echo`.

Bisogna ricordare che l'utente di default del webserver si chiama **www-data** e non ha il permesso di mandare in esecuzione dei file nemmeno con il comando `sudo`. Pertanto se non è stato fatto, bisogna modificare il file `sudoers` presente nella cartella `/etc`, aggiungendo in coda quanto segue:

www-data ALL=NOPASSWD: ALL

Per editare questo file si può digitare da CLI il comando `sudo visudo`, o `sudo nano /etc/sudoers`

3) File `sensori.py` presente nel Server Raspberry al percorso `/home/pi/Desktop`

Questo file contiene il programma che verrà eseguito per leggere il valore del sensore richiesto.

```
#importo il modulo sys per poter ricevere e restituire
#parametri da riga di comando
import sys
nome_script, richiesta, rosso, verde, blu = sys.argv

from time import sleep      #importo l'oggetto sleep dal modulo time
from sense_hat import SenseHat #importo l'oggetto SenseHat dal modulo sense_hat
sense = SenseHat()         #creo l'oggetto sense

sense.clear()              #azzerò il display

black = (0,0,0)           #creo la variabile con i 3 colori per il nero

#converto le stringhe contenente i 3 colori in intero
val1=int(rosso)
val2=int(verde)
val3=int(blu)

color=(val1,val2,val3)     #creo la variabile color con i colori richiesti

#controllo il sensore richiesto
if richiesta=="P":
    press=sense.get_pressure() #leggo la pressione
    val=round(press,2)         #arrotondo 2 cifre dopo la virgola
    message="Pressione="+str(val)+"mBar" #preparo il messaggio da visualizzare
    print("Pressione=")       #scrivo a video la scritta
    print(val)                #scrivo a video il contenuto della variabile
    print("mBar")             #scrivo a video l'unita' di misura
elif richiesta=="T":
    temp=sense.get_temperature() #leggo la temperatura
    val=round(temp,2)           #arrotondo, 2 cifre dopo la virgola
    message="Temperatura="+str(val)+" C" #preparo il messaggio da visualizzare
    print("Temperatura=")      #scrivo a video la scritta
    print(val)                 #scrivo a video il contenuto della variabile
    print(" C")                #scrivo a video l'unita' di misura
elif richiesta=="U":
    umid=sense.get_humidity()   #leggo umidità
    val=round(umid,2)          #arrotondo, 2 cifre dopo la virgola
    message="Umidità="+str(val)+"%" #preparo il messaggio da visualizzare
    print("Umidità=")         #scrivo a video la scritta
    print(val)                #scrivo a video il contenuto della variabile
    print("%")                #scrivo a video l'unita' di misura
else:
    message="Nessuna richiesta"
    val=0
    print(message)
    print(val)
    print(val)

sense.show_message(message,text_colour=color,back_colour=black,scroll_speed=0.05)
```

Il file è ben commentato ma proviamo a descrivere le parti che lo compongono.

Nella parte iniziale vengono importati i moduli necessari per il programma.

Il modulo **sys** consente di passare dei comandi da riga di comando.

Digitando ad esempio **python sensori.py Temperatura 0 255 0** viene lanciato il programma e vengono passate 4 variabili di tipo stringa indicate e cioè: "Temperatura" "0" "255" "0".

Il modulo **sense_hat** permette invece di creare l'oggetto **SenseHat** chiamato **sense**, che ci permetterà tramite i suoi metodi di acquisire il valore delle grandezze richieste.

Il modulo **time**, non è utilizzato nel programma ma serve per generare eventuali ritardi, **sleep**.

Successivamente all'importazione dei moduli, vengono create delle variabili per gestire il colore dello sfondo e della scritta. La variabile che rappresenta un colore, viene creata partendo dai 3 colori fondamentali Rosso, Verde e Blu, nel caso della variabile **black**, i 3 valori valgono 0, **black=(0,0,0)**.

La variabile **color**, invece dipende dai valori delle 3 variabili **rosso**, **verde** e **blu**.

I 3 valori sono stati importati all'inizio insieme alla variabile **richiesta**, con la funzione **argv**.

La sintassi è la seguente: **nome_script, richiesta, rosso, verde, blu = sys.argv**

Nel nostro caso digitando la stringa di esempio sopra, avremo che nella variabile **richiesta** andrà la stringa "Temperatura" ed in ordine nelle altre variabili andranno **rosso="0"** **verde="255"** e **blu="0"**.

I valori dei 3 colori sono però rappresentati come stringa e dovranno essere convertiti in numeri interi con le 3 assegnazioni: **val1=int(rosso)** **val2=int(verde)** e **val3=int(blu)**.

Successivamente verrà creata la variabile **color** con i 3 valori contenuti in **val1**, **val2** e **val3**.

Nella terza parte con una serie di if ed elif, si verifica il contenuto della variabile **richiesta**, ed in base ad esso si utilizzerà la funzione **sense.get** per acquisire uno dei 3 valori.

Il valore viene arrotondato a due cifre dopo la virgola con la funzione **round**, ed inserito nella stringa **message** insieme al nome della relativa grandezza e della sua unità di misura.

Successivamente vengono fatte 3 scritture a video, **print**, che restituiscono 3 stringhe al file php, che ha lanciato il file python.

Con l'ultima riga viene eseguita la funzione **sense.show_message** per visualizzare sul display a matrice di led la stringa contenuta in **message**.

CONCLUSIONI

Questo è uno dei modi più semplici per interagire a distanza con la scheda Raspberry, e partendo da questo esempio si possono strutturare altri semplici progetti.

Bisogna inoltre ricordare che il file html, non necessariamente deve essere presente sul PC client, ma può risiedere anche nello stesso server nella stessa cartella del file php.

In questo caso potremo accedere al file digitando sulla barra di indirizzi del browser, l'indirizzo ip seguito dal nome del file html, es 192.168.1.113/prova.html.