

Matteo Baccan

Realizzazione di una stazione meteo

Quali prodotti software sono stati utilizzati per la realizzazione della stazione meteo? In questo breve talk vedremo le scelte che sono state fatte per poter avere una GUI responsive in ambiente Raspberry Pi

matteo@baccan.it – <http://www.baccan.it>



Matteo Baccan

Divulgatore informatico, giornalista per riviste tecniche nazionali oltre 700 articoli per: Dev, Login, Computer Programming, Mokabyte. Relatore tecnico ad eventi di programmazione: Borland Forum 2000, Webbit 2004, JIP day 2005, Javaday (2006,2007,2010) PHPDay (2008, 2010), CONFSL 2010, WebTech 2010, Codemotin (2011,2012,2013,2014), Vodafone Digital Day (2014). Autore di Corso di C# ISBN 8881500167. Autore di JobCrawler e HTML2POP3 (Oltre 900.000 download su SourceForge)

WEB - Architettura LAMP

A causa della potenza limitata di Raspberry Pi, è stato scelto di utilizzare uno stack LAMP per la realizzazione della componente Web

- Linux - Raspbian
- Apache
- MySQL
- PHP

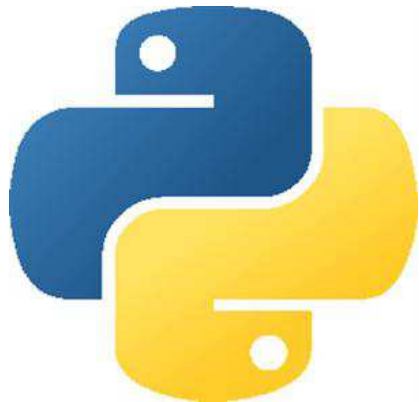


Sensori - Python

Alcuni dei sensori utilizzati sono forniti da Adafruit Industries.

Oltre a fornire i sensori, Adafruit fornisce anche alcuni esempi Python di come leggere dei dati rilevati dai sensori stessi.

Per questo motivo, abbiamo optato per una personalizzazione di tali programmi, in modo che non si limitassero a leggere i dati e a visualizzarli, ma che li andassero anche a scrivere su un database MySQL.



WEB - GUI

Per la realizzazione della GUI web sono stati utilizzati alcuni componenti grafici, ideali per la realizzazione di GUI responsive e in linea con gli attuali standard di mercato

- Bootstrap – per la struttura delle pagine



- JQuery – per il codice aggiuntivo che dinamicamente aggiorna le pagine



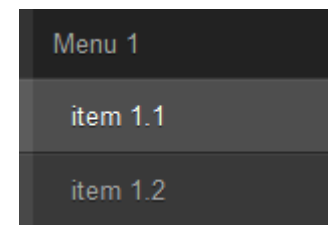
- Flot – per la realizzazione dei grafici



- Font Awesome – per le icone personalizzate



- MetisMenu – per i menu del programma



Funzionamento

Alla partenza di Raspberry viene invocato il seguente service

```
'start')
    #!/bin/sh

    service apache2 start
    service mysql start
    sudo /home/pi/meteo/Adafruit_BMP085_mysql.py &

    exit 0
    touch /var/lock/subsys/meteo
    ;;

'stop')
    #!/bin/sh

    sudo kill $(ps aux | grep 'Adafruit_BMP085_mysql.py' | awk '{print $2}')
    service apache2 stop
    service mysql stop

    exit 0
    rm -f /var/lock/subsys/meteo
    ;;

*)
    echo "Usage: $0 { start | stop }"
    ;;
```

Lettura dei sensori



```
# Initialise the BMP085 and use STANDARD mode (default value)
# bmp = BMP085(0x77, debug=True)
bmp = BMP085(0x77)

# To specify a different operating mode, uncomment one of the following:
# bmp = BMP085(0x77, 0) # ULTRALOWPOWER Mode
# bmp = BMP085(0x77, 1) # STANDARD Mode
# bmp = BMP085(0x77, 2) # HIRES Mode
# bmp = BMP085(0x77, 3) # ULTRAHIRES Mode

temp = bmp.readTemperature()

# Read the current barometric pressure level
pressure = bmp.readPressure()

# To calculate altitude based on an estimated mean sea level pressure
# (1013.25 hPa) call the function as follows, but this won't be very accurate
altitude = bmp.readAltitude()
```

Scrittura su MySQL

```
while 1==1:
insert = "insert into sensori(stazione,data,temperatura,pressione,altitudine)
values('Linux','%s',%.2f,%.2f,%.2f);" % (datetime.datetime.now().strftime("%Y-%m-
%d %H:%M:%S"), temp , (pressure / 100.0) , altitude)

con = 0
try:
    con = mdb.connect('192.168.2.22', 'pi', 'raspberrypi', 'meteo')
    cur = con.cursor()
    cur.execute( insert )

except _mysql.Error, e:
    print "Error %d: %s" % (e.args[0], e.args[1])
    sys.exit(1)

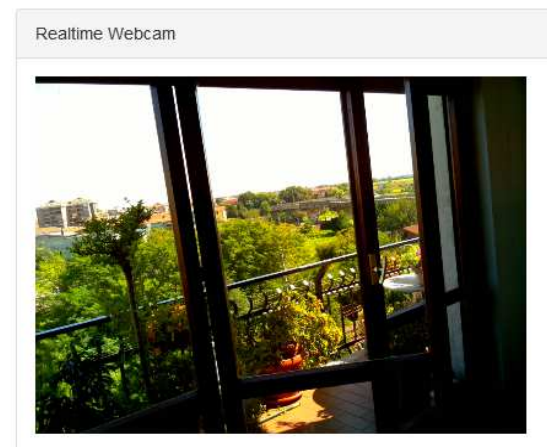
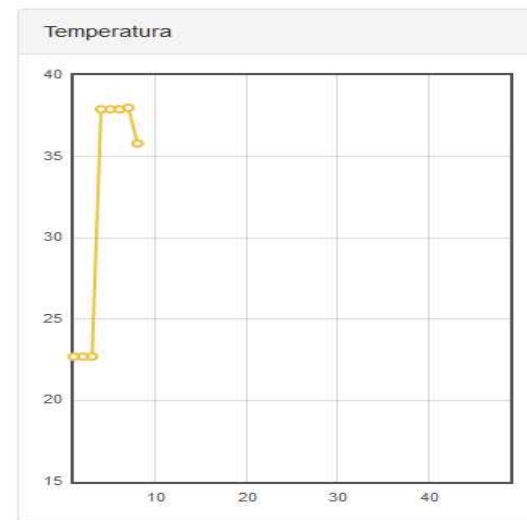
finally:
    if con:
        con.close()

time.sleep(1);
```


Componenti - GUI

- Visualizzazione sensori
 - Grafici Flot aggiornati realtime tramite un webservice HTTP/JSON interrogato tramite JQuery

- TimeLapsed
 - In questo caso, lettura tramite PHP della cartella timelapsed e visualizzazione di uno slideshow delle immagini catturate



DEMO

DOMANDE?





Matteo Baccan

matteo@baccan.it

<http://www.baccan.it>