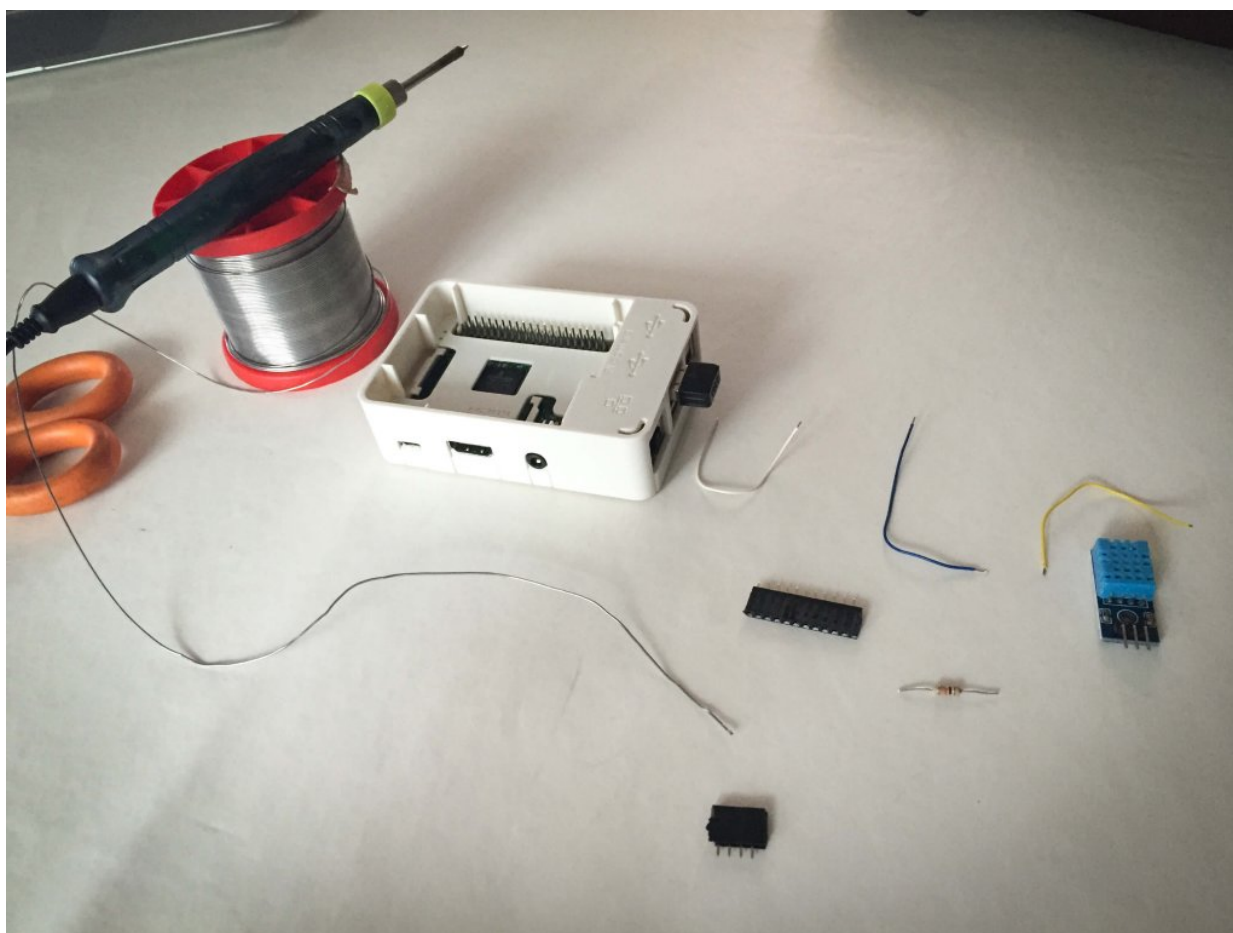


IL SENSORE DHT11 SU RASPBERRY PI2 & LOGGER SU GDOCS

18 agosto 2015 , di Francesco Cozzi , Commenti 3

Categoria:Raspberry Pi



Il DHT11 è un sensore digitale low-cost di temperatura e umidità, che fa uso di un sensore capacitativo di umidità e un termistore per misurare l'aria circostante, e fa uscire un segnale digitale sul pin data (pin centrale).

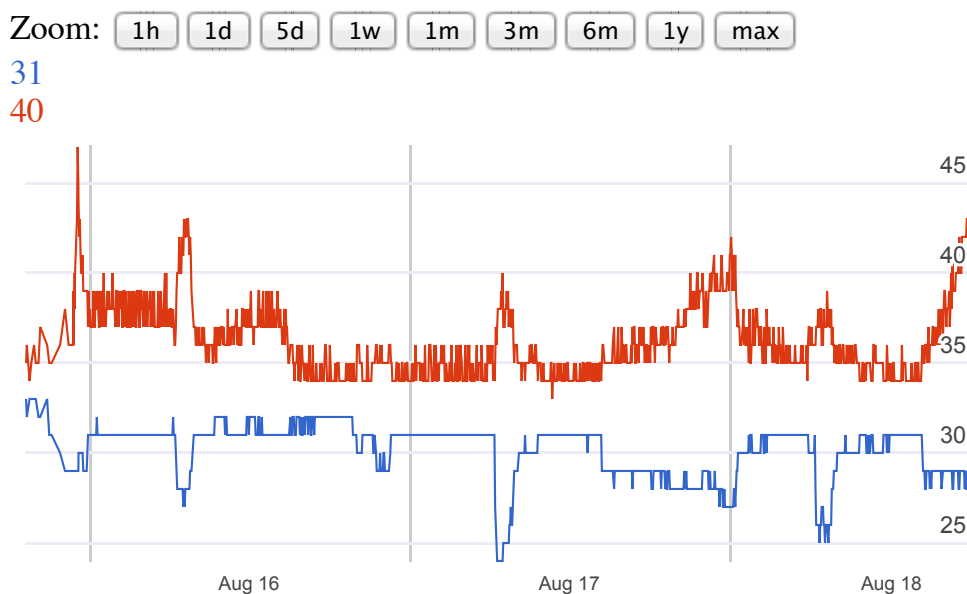
In questo articolo spiego come ho usato la libreria DHT Python ufficiale di Adafruit che usa il linguaggio C per sfruttare l'alta velocità del polling GPIO della Pi per gestire l'output in bit-banged (comunicazione seriale) del sensore. Molti sensori low cost possiedono un output con formattazioni

inusuali, e in questo caso, una “codifica Manchester” in output, che non è SPI, I2C o 1-Wire, deve essere gestita in continuazione dalla Pi ed essere decodificata. Per fortuna, le librerie C GPIO sono veloci abbastanza per decodificare l'output.

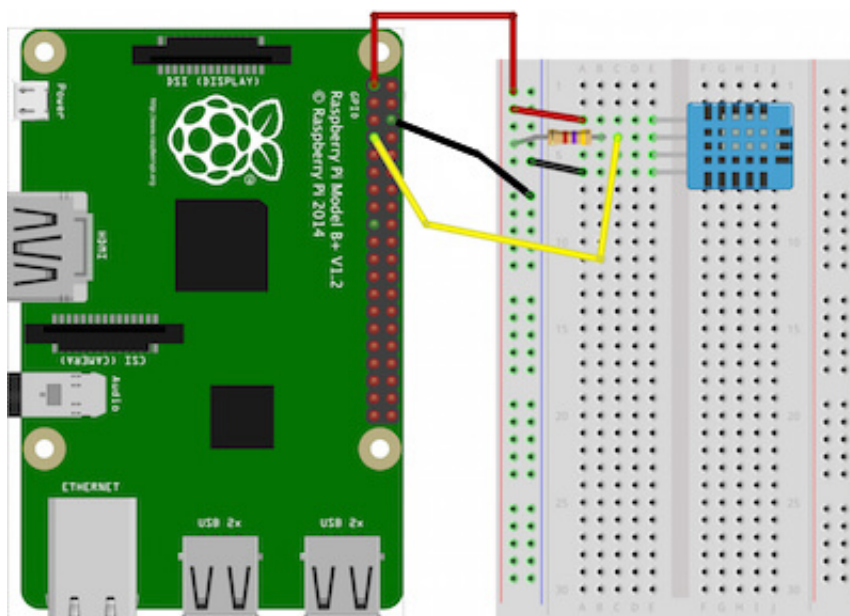
[AMAZONPRODUCTS asin="B00WMNBK2K"]

Successivamente, sfruttando la potenza di Python, ho aggiunto la possibilità di utilizzare un foglio di calcolo di Google Docs come sistema di registrazione dei dati ogni tre minuti.

Questo è il risultato finale in tempo reale del sensore a casa mia:



Ok, vediamo, procediamo con il cablaggio del sensore sulla Raspberry Pi2 (sarà lo stesso su una Raspberry Pi e, con differenti pin anche su una Beaglebone Black).



[AMAZONPRODUCTS asin="B00T2U7R7I"]

Raspberry Pi GPIO DHT11

3.3V Pin1

VCC Pin1

GND Pin6

GND Pin4

GPIO4 Pin7

Data Pin2

È importante mettere una resistenza da 4,7 o 10 Kohm tra il pin data e pin VCC per far salire il segnale. Prova prima con 4.7K e se non funziona utilizzare un resistore da 10K.

È possibile utilizzare qualsiasi pin GPIO per il pin data del sensore DHT, ma in questo articolo userò il GPIO4 Pin 7 nel codice.

I seguenti comandi e le istruzioni si basano sulla distribuzione Raspbian su Raspberry Pi2.

Ho usato un codice davvero ben scritto in Python e C da Adafruit per lavorare con sensori DHT ed è disponibile su Github al link: https://github.com/adafruit/Adafruit_Python_DHT.

Usare Git per clonare il software direttamente sul Raspberry utilizzando il terminale:

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
```

Installa alcune dipendenze su Rasbian o Debian:

```
sudo apt-get update
sudo apt-get install build-essential python-dev python-openssl
```

Vai avanti se si ottiene qualche errore o se si sono già installati alcuni pacchetti.

Ora, per installare la libreria eseguire:

```
sudo python setup.py install
```

Ora è possibile testare la libreria con qualche esempio, navigare nella cartella examples:

```
cd examples
```

Ed eseguire come root questo script Python di esempio, dove 11 sta per sensore DHT11 e 4 per pin GPIO4:

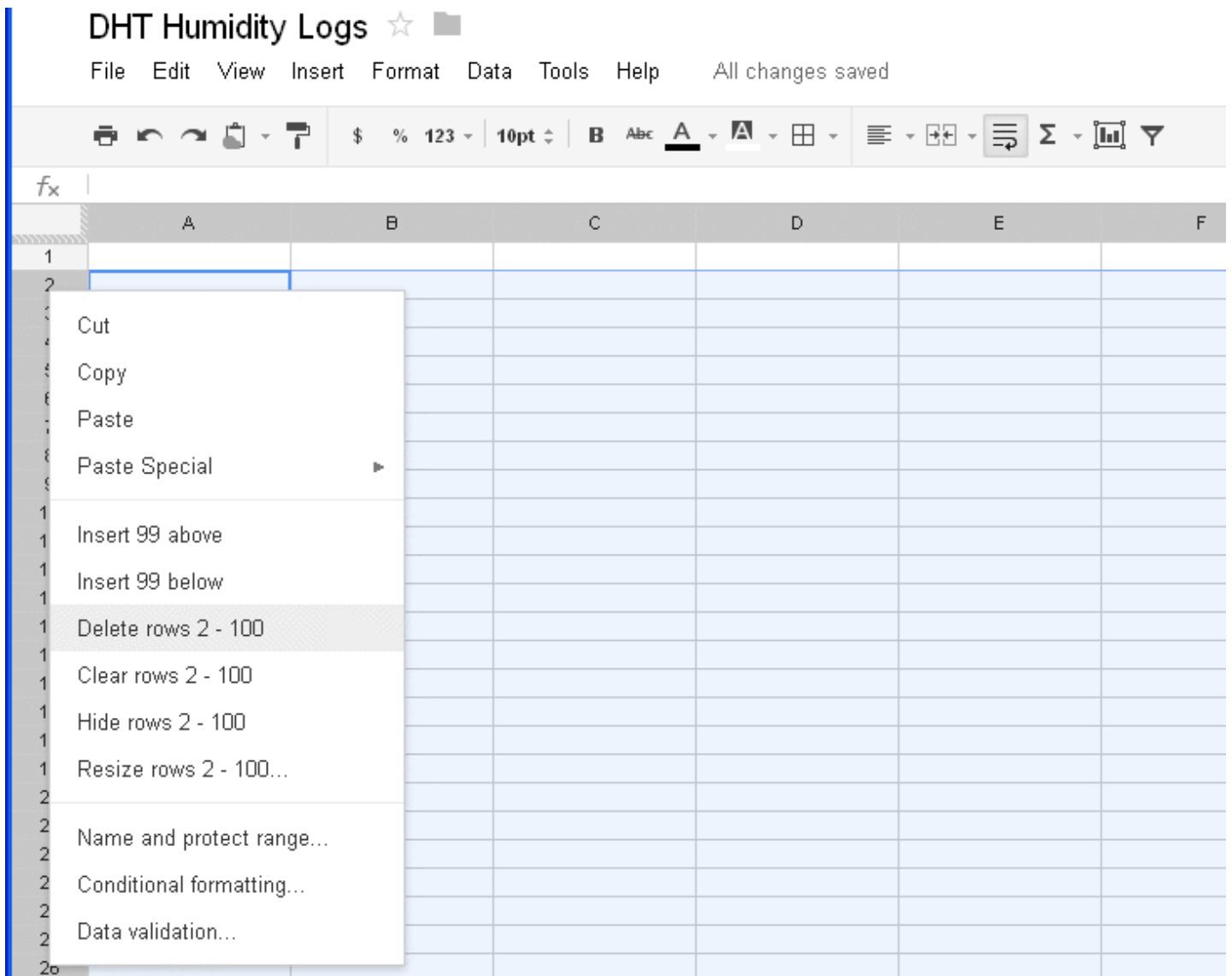
```
sudo ./AdafruitDHT.py 11 4
```

Si dovrebbe ottenere il valore di temperatura e umidità visualizzato come di seguito:

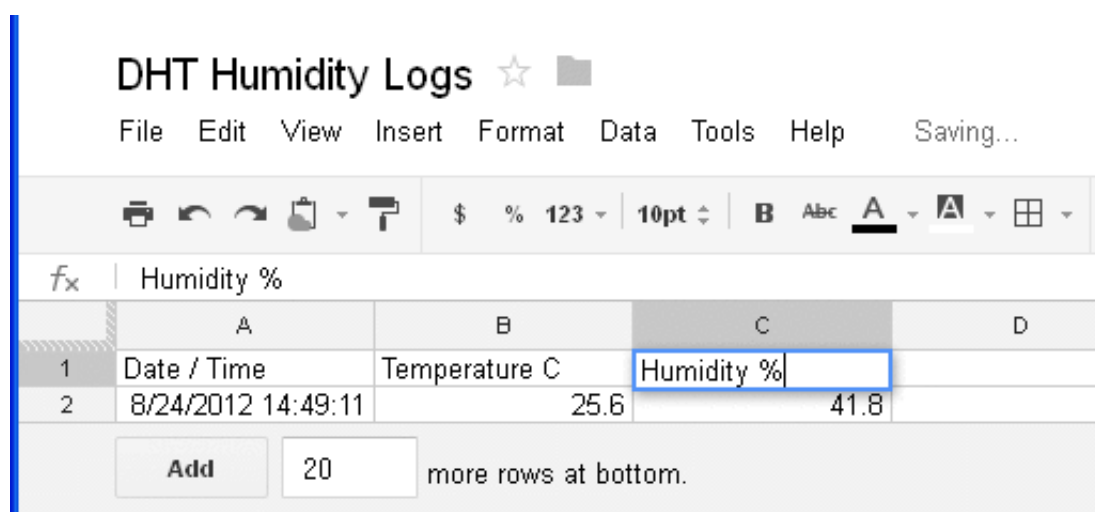
```
pi@raspberrypi /home/Adafruit_Python_DHT/source $ cd /homeAdafruit_P
ython_DHT/examples
pi@raspberrypi /home/Adafruit_Python_DHT/examples $ sudo ./AdafruitD
HT.py 11 4
Temp=28.0*C Humidity=38.0%
```

Il mio progetto consiste nella registrazione dei valori di temperatura e umidità ogni tre minuti su un foglio di calcolo di Google Docs per ottenere un grafico in tempo reale.

Quindi prepara il foglio di calcolo per il logging in GDocs, creare il foglio di calcolo ed eliminare tutte le righe, ma non il primo:



Nella prima riga rimasta scrivere l'intestazione:



Ora dobbiamo collegare la Raspberry Pi al foglio di calcolo GDocs e useremo la libreria Python **gsread** e il metodo di accesso **OAuth2**.

Questi passaggi sono più complessi e sono basati sul nuovo metodo di autenticazione introdotto da Google da Aprile 2015.

Per ottenere le credenziali OAuth2 seguire i passi sul sito ufficiale della libreria Python gspread:

- [gspread – Using OAuth2 for Authorization](#)

Dopo aver eseguito le procedure nel documento di cui sopra si dovrebbe aver scaricato un file .json, come SpreadsheetName- (codice) .json.

Mettere questo file .json nella stessa cartella come l'esempio google_spreadsheet.py. Se non si inserisce questo file nella stessa cartella allora l'autenticazione avrà esito negativo e non sarai in grado di aggiornare il foglio di calcolo!

Un ultimo passo che **deve essere completato** è quello di condividere il foglio di calcolo di Google con l'indirizzo email associato alle credenziali OAuth2. Aprire il file .json e cercare il **“client_email”**: la linea che assomiglia a questa (ma con un indirizzo email diverso):

```
"client_email": "149345334675-md0qff5f0kib41meu20f7d1habos3qcu@devel  
oper.gserviceaccount.com",
```

Usando la voce del menu **File -> Condividi ...** condividi il foglio di calcolo sia in lettura che scrittura all'indirizzo e-mail trovato sopra. Assicurati di condividere il tuo foglio di calcolo o non sarai in grado di aggiornarlo con lo script!

Ora installare la libreria gspread e il cliente OAuth2:

```
sudo apt-get update
sudo apt-get install python-pip
sudo pip install gspread oauth2client
```

Quindi aprire lo script Python **google_spreadsheet.py** nella cartella examples e compilare la configurazione corretta (tipo di sensore DHT, Pin, il nome del file JSON, nome del file GDocs):

```
# Type of sensor, can be Adafruit_DHT.DHT11, Adafruit_DHT.DHT22, or
Adafruit_DHT.AM2302.
DHT_TYPE = Adafruit_DHT.DHT11

# Example of sensor connected to Raspberry Pi pin 23
DHT_PIN = 4
# Example of sensor connected to Beaglebone Black pin P8_11
#DHT_PIN = 'P8_11'

# Google Docs OAuth credential JSON file. Note that the process for
authenticating
# ...
GDOCS_OAUTH_JSON = 'your SpreadsheetData-*.json file name'

# Google Docs spreadsheet name.
GDOCS_SPREADSHEET_NAME = 'your google docs spreadsheet name'
```

La frequenza di misura può essere regolata modificando la configurazione **FREQUENCY_SECONDS** nel codice Python, io ho impostato 180 secondi, 3 minuti, perché impostando cinque minuti o più, dopo due misure avveniva un errore di timeout, che sto ancora indagando, forse dovuto a troppa latenza tra una misura e l'altra.

Ora eseguire lo script Python:

[← INDIETRO](#)

[MENU](#)

Dovreste vedere i valori popolare le righe del foglio di calcolo. Con questi dati ho costruito un grafico temporale dei valori di umidità e temperatura come si può vedere all'inizio di questo articolo.

L'evoluzione di questo progetto sarà quello di aggiungere un sensore per misurare la velocità del vento e un sensore di pressione atmosferica, tutto sarà collocato all'esterno e funzionerà come stazione meteo.

Tags: adafruit, Beaglebone Black, dht, dht11, gdocs, google docs, gpio, humidity, python, raspberry, raspberry pi, raspberry pi2, spreadsheet, temperature

COMMENTI (3)

Riccardo

12 giugno 2017

Ho seguito tutta la procedura e tutto funziona alla perfezione, quindi già ti ringrazio!

Ho però un grosso problema che non riesco a risolvere: come faccio a creare il grafico una volta creato il file con tutti i valori? (come quello da te pubblicato per intenderci)

Grazie!

[Rispondi](#)



Francesco Cozzi

16 giugno 2017

Ciao Riccardo,

mi fa piacere che la mia guida ti sia risultata chiara. Per quanto riguarda il grafico è piuttosto facile, in Google Fogli vai su Inserisci > Grafico e come sorgente dei dati gli dai le colonne che hai raccolto dal sensore.

[Rispondi](#)



Riccardo

17 giugno 2017

Ho provato in tutti i modi ma proprio non riesco, che tipo di grafico hai utilizzato? devo selezionare tutte e 3 le colonne intestazioni comprese? Scusa ancora per il disturbo!

[Rispondi](#)



LASCIA UN COMMENTO

//

Nome

Email

Sito Web

Commento all'articolo

- ☐ Avvertimi via email in caso di risposte al mio commento.
- ☐ Avvertimi via email alla pubblicazione di un nuovo articolo

Cerca

CATEGORIE

Blog
HamRadio
Marketing
Raspberry Pi
WordPress

COMMENTI RECENTI

Riccardo su Il sensore DHT11 su Raspberry Pi2 & Logger su GDocs

Francesco Cozzi su Il sensore DHT11 su Raspberry Pi2 & Logger su GDocs

Riccardo su Il sensore DHT11 su Raspberry Pi2 & Logger su GDocs

Francesco Cozzi su Piattaforma intranet con WordPress

Alessandro Nizzo su Piattaforma intranet con WordPress

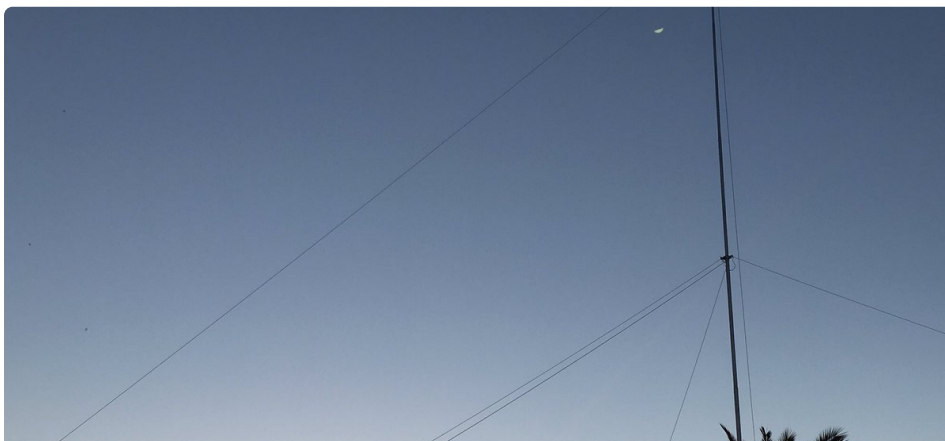
SEGUIMI SU TWITTER

Tweet di @IZ7KHR



Francesco Cozzi @IZ7KHR

Sunrise @ IT9YVO/IO9A Contest Station for @rsgbiota
contest #hamradio

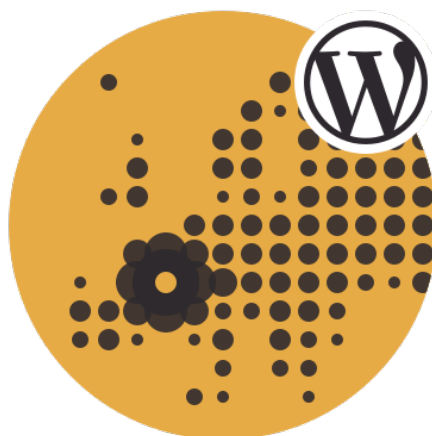


Incorpora

Visualizza su Twitter

Successivo

VOLUNTEER @ WORDCAMP EUROPE 2017



WORDCAMP EUROPE 2017

June 15-17, Paris, France | #WCEU

ORGANIZER @ WORDPRESS MEETUP BARI



CATEGORIE

[Blog](#)
[HamRadio](#)
[Marketing](#)
[Raspberry Pi](#)
[WordPress](#)

COMMENTI RECENTI

Riccardo su [Il sensore DHT11 su Raspberry Pi2 & Logger su GDocs](#)
Francesco Cozzi su [Il sensore DHT11 su Raspberry Pi2 & Logger su GDocs](#)
Riccardo su [Il sensore DHT11 su Raspberry Pi2 & Logger su GDocs](#)
Francesco Cozzi su [Piattaforma intranet con WordPress](#)
Alessandro Nizzo su [Piattaforma intranet con WordPress](#)

MI TROVI QUI:



CERCA

Cerca

IL MIO HOSTING È DIFFERENTE



©2016 BY FRANCESCO COZZI - PROUDLY POWERED BY WORDPRESS

[SITEMAP](#)