

Nello snippet [precedente](#) , avevo scritto come installare, su una Debian, l'IDE ufficiale di Arduino. In questa mattinata vedremo un modo per non utilizzarlo affatto XD

In altre parole, mentre l'IDE puo' essere un rapido modo per iniziare a lavorare con Arduino, l'utente Linux piu' smaliziato potrebbe voler chiedere strumenti piu' potenti, con cui magari ha gia' lavorato in passato. In questa guida vedremo come utilizzare ViM come editor di testo per scrivere il codice. Makefile per la compilazione e l'upload del codice su Arduino, e un semplice script in python per la lettura della porta seriale.

WARNING: L'obiettivo di questa guida non e' di iniziarvi all'uso di tali strumenti, quando il permettervi di utilizzarli in modo proficuo con Arduino!

- tale guida e' stata testata, da me, sulla solita Debian Wheezy, con Arduino 2009 munito di microprocessore Atmega328
- i file ritenuti utili li trovate a questo indirizzo [https://github.com/GrayHats/arduino\\_scripts](https://github.com/GrayHats/arduino_scripts) (e' un repository git che terro' aggiornato man mano che aumentera' la mia esperienza con Arduino XD)

## L'Editor di testo - ViM

Per prima cosa il nostro ViM necessita di un plugin per evidenziare correttamente la sintassi del linguaggio utilizzato per programmare Arduino. Scaricate e installate [questo file](#)

In alternativa, se avete già scaricato tutti gli script dal nostro repository git vi basterà digitare:

```
# siamo già nella cartella arduino_scripts XD
mkdir -p ~/.vim/syntax/
cp utility/arduino.vim ~/.vim/syntax/
echo "autocmd! BufNewFile,BufRead *.pde setlocal ft=arduino" >> ~/.vimrc
```

## Creare/Compilare/Uploadare Sketch

**REGOLA #1 - Ogni sketch vuole una cartella dedicata.**

Quindi niente cartelle contenenti milioni di sketch... L'IDE di arduino fa questo in modo nascosto a voi, voi lo farete in modo consapevole. Punto.

**REGOLA #2 - In ogni cartella va copiato il Makefile. Tale Makefile va modificato in base all'hw arduino che avete.**

Ad esempio, nel repository git mio trovate il **Makefile** che funziona con Arduino 2009 con processore atmega328. Un modo per ottenere i valori che vi servono è consultare il file **utility/boards.txt**

La sintassi da utilizzare è la seguente

```
$ ls -l # solo per farvi vedere cosa è presente nella cartella
Makefile
temperature_sensor_lm35.pde
```

```
$ make TARGET=temperature_sensor_lm35
```

A questo punto, se il codice compila senza errori vi ritroverete con questa situazione

```
$ ls -1 build-cli Makefile temperature_sensor_lm35.pde
```

e potete upoloadare il file su arduino con

```
$ make TARGET=temperature_sensor_lm35 upload
```

Per ricompilare nuovamente il codice e' necessario eliminare la cartella **build-cli**

## Serial Monitor

Qualora vi servisse emulare questa funzione dell'editor, ho preparato per voi un banale script in python: **read\_serial.py**

Fine, per ora. Buona programmazione.

## Recipe - GNU/Linux, Arduino, ViM e Makefile

Scritto da {ga=allanon}

---

Se avete suggerimenti, script e materiale, fatecelo sapere, o qui o postate nel nostro forum, o postatelo [qui](#)