

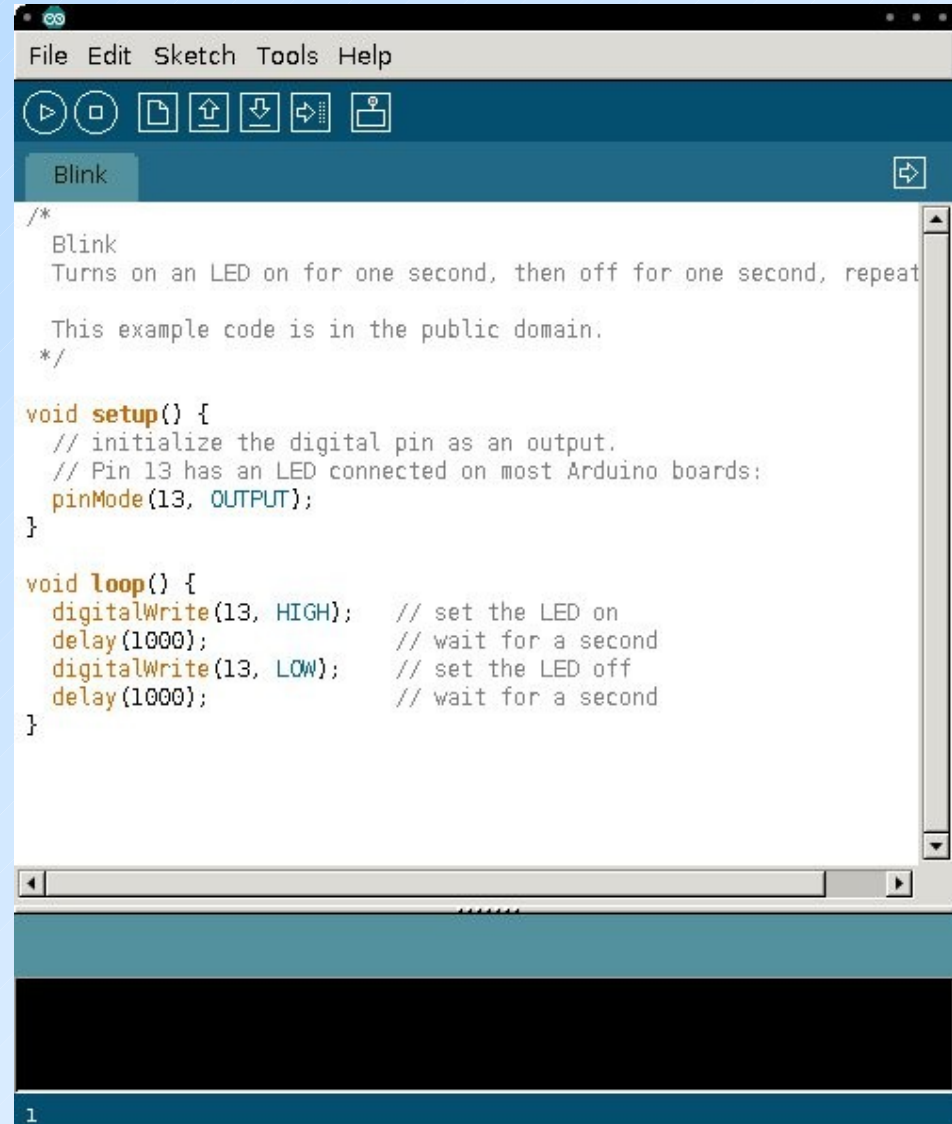
ARDUINO MOTORI e PWM

Come usare arduino per gestire i motori elettrici



Quali programmi?

- Arduino

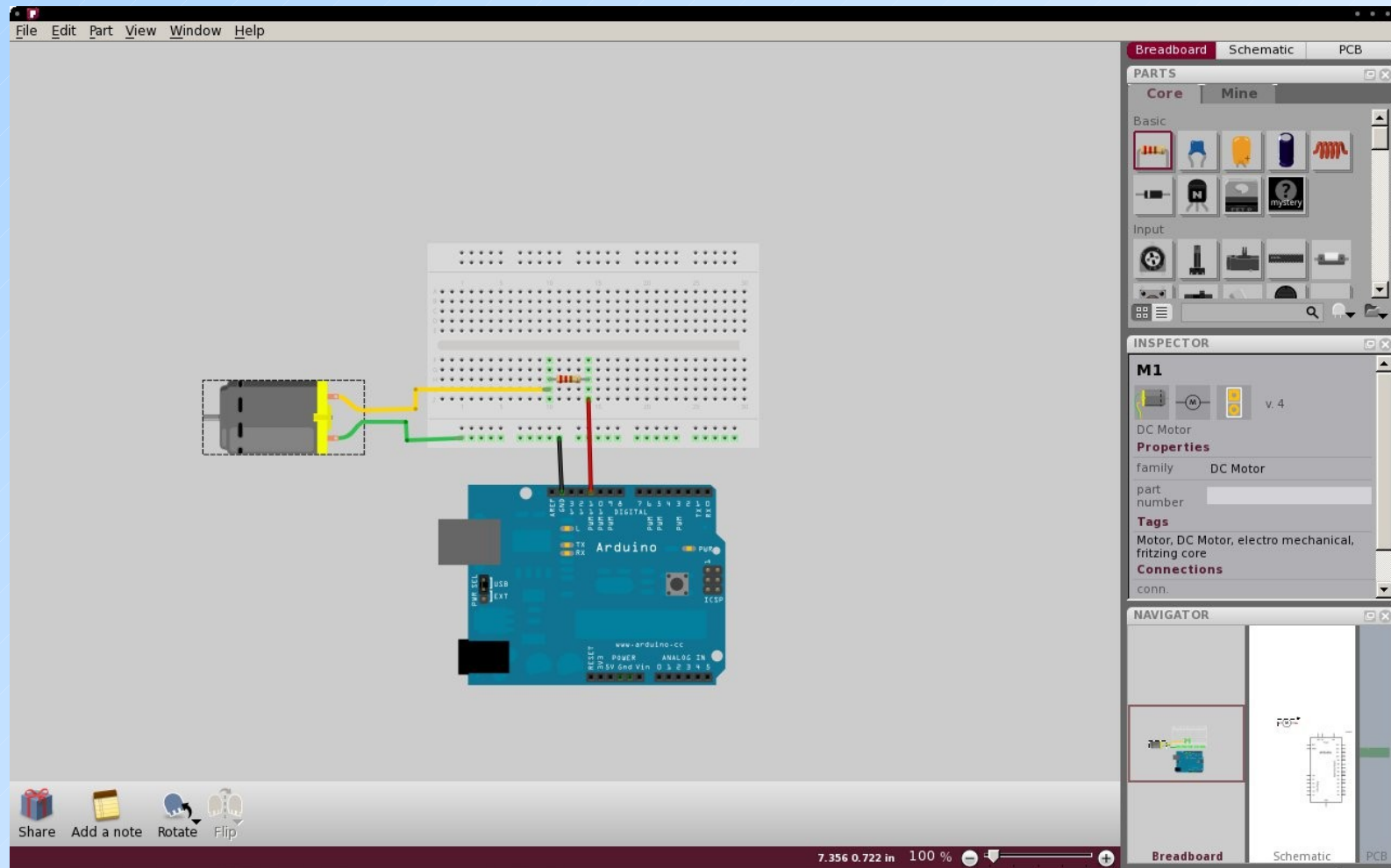
A screenshot of the Arduino IDE interface. The window title is "Arduino IDE". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for running, stopping, saving, opening, and other sketch-related actions. The "Sketch" menu is open, showing a list of sketches, with "Blink" selected. The main text area displays the code for the "Blink" sketch. The code is as follows:

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeat  
  
  This example code is in the public domain.  
  */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);            // wait for a second  
  digitalWrite(13, LOW);  // set the LED off  
  delay(1000);            // wait for a second  
}
```

The status bar at the bottom shows the line number "1".

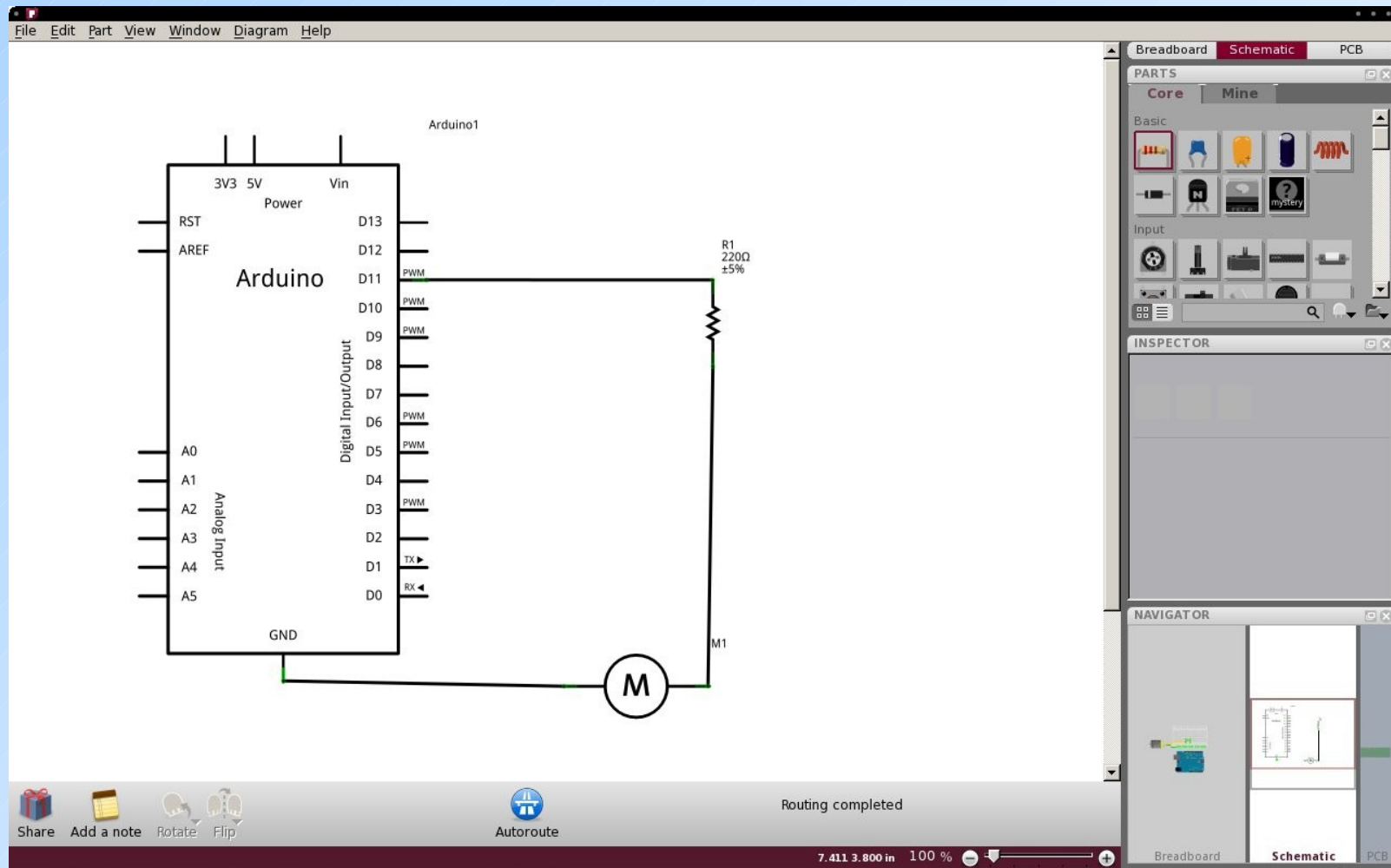
Quali programmi?

- Fritzing



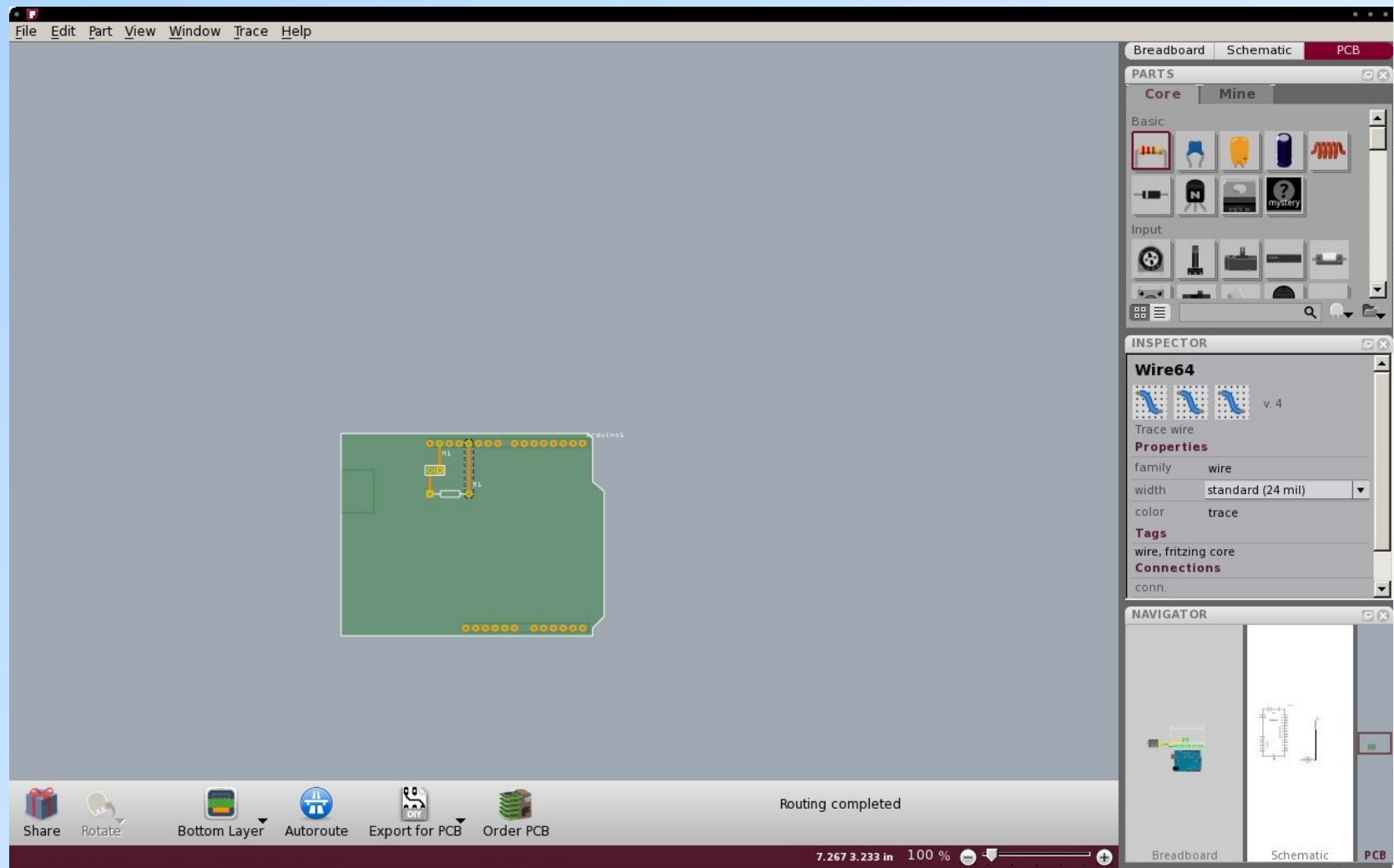
Quali programmi?

- Fritzing



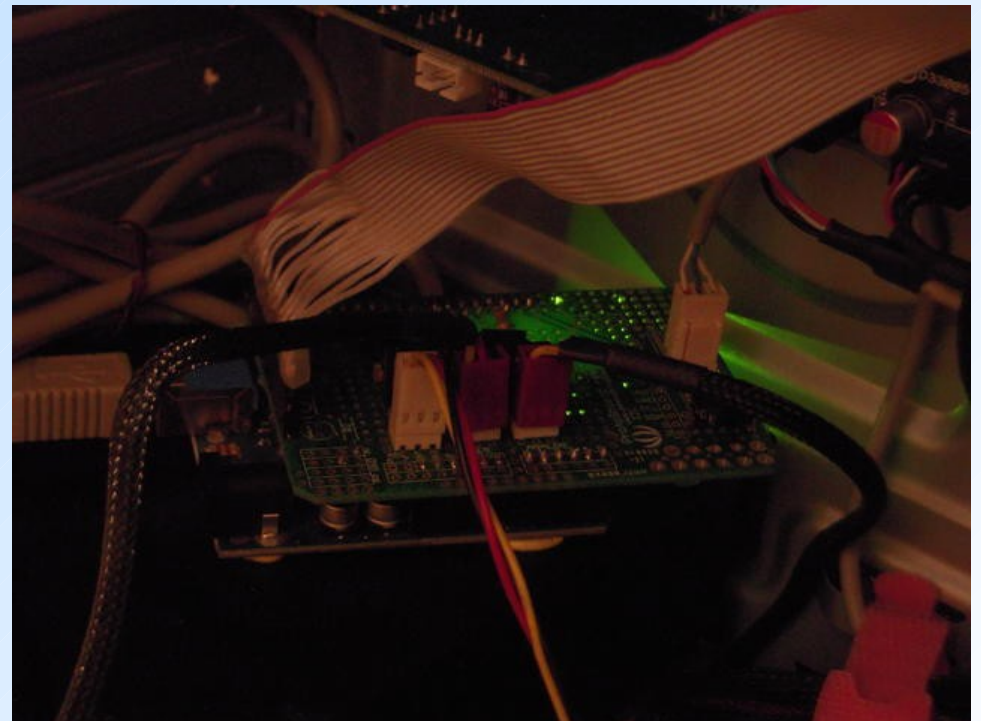
Quali programmi?

- Fritzing



Cosa e perchè

- Vedremo come costruire un controller per ventole.



Caratteristiche del sistema

- Temperature inviate dal SO (molto più preciso dei sensori esterni)
- È arduino a decidere la velocità delle ventole, in base alle temperature

Caratteristiche del sistema

- Comunicazione con il SO tramite USB, e conseguente alimentazione
- Protocollo di comunicazione estremamente semplice e banale

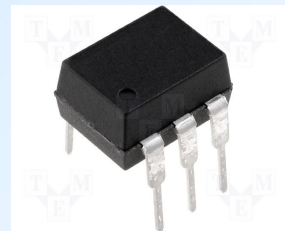
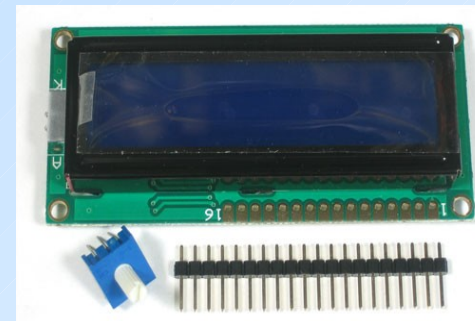
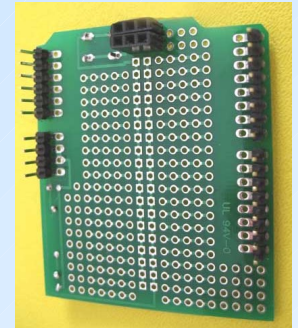
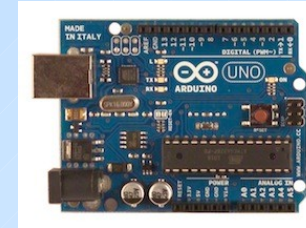
Caratteristiche del sistema

- Temperature e velocità ventole vengono mostrate a schermo (lcd)

La lista della spesa

- Arduino (20-30 €)
- Shield generica/millefori (5€)
- Lcd 16x2 (10€)
- CNY75 (optoisolatore)
- BD137 (transistor)
- Resistenze (550 Ω , 10 K Ω)
- Condensatori (0.5-10 μ F)
- Connettore per ventole
- Diodi

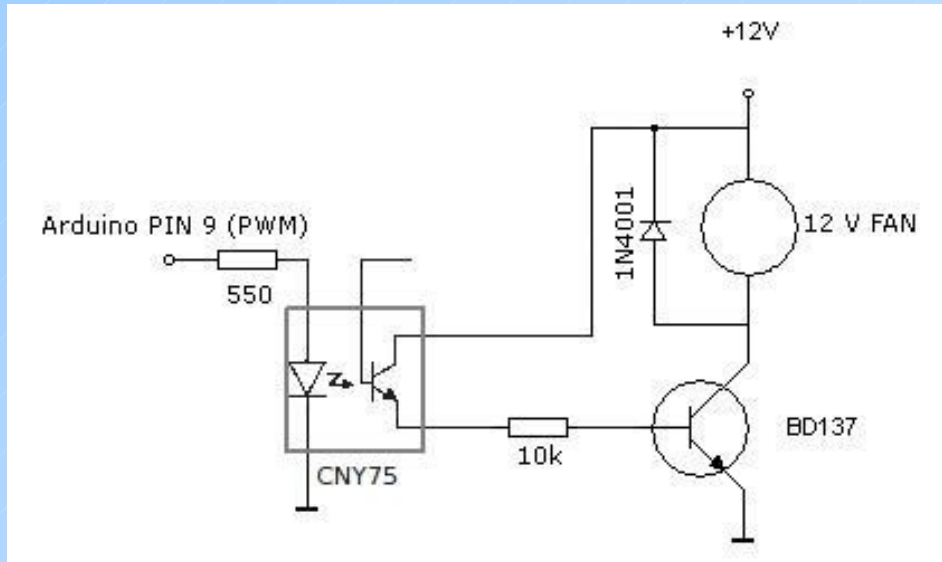
In totale si spende intorno ai 50€



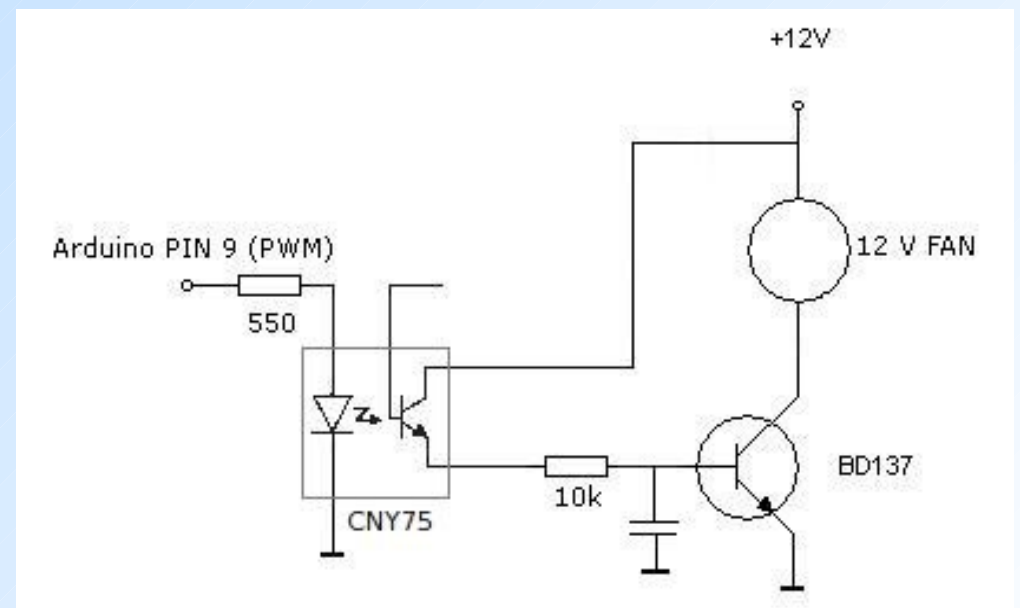
Il circuito

circuito originale

<http://www.uchobby.com/index.php/2007/09/23/arduino-temperature-controlled-pc-fan/>

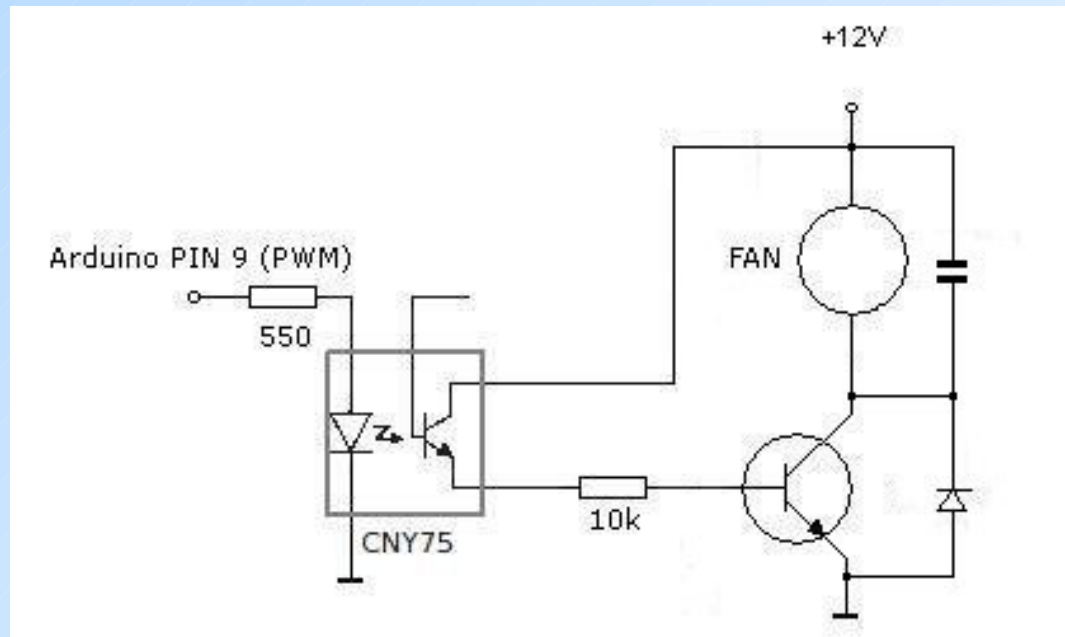


circuito modificato



Note sul circuito

- Vi serve un circuito per ogni ventola
- Il condensatore è stato aggiunto successivamente per evitare fastidiosi fischi delle ventole, vi conviene ripensare il circuito per mettere un filtro in zona più idonea, tipo:



Il software

- All'avvio le ventole partono ad una velocità preimpostata (questa era l'idea)

http://research.google.com/archive/disk_failures.pdf

Il software

- Arduino invia al SO un segnale (char) ogni volta che vuole le temperature, il SO risponde con 3 temperature (char): processore, chipset e harddisk.
- Se il SO non risponde entro un certo tempo, reinvia la richiesta

Calcolare la velocità

- Volevo che la velocità delle ventole dipendesse sia dai componenti che raffreddano direttamente, sia dalla cpu

Calcolare la velocità

- Ho usato questa formula: $\frac{tempmb^2}{mbconst} + \frac{tempcpu^2}{cpuconst}$
dove
mbconst=9
cpuconst=100
- Similmente per l'hd

Codice Arduino

Codice (PC)

- Scritto in C
- Temperature prese da libsensors e hddtemp

Note e ipotetiche migliorie

- Scegliere il condensatore
- Usare il tip-120
- Modificare il clock dei pwm
- Intercettare i kill (TERM)

Domande?

GRAZIE!

Marco Impallaria

<http://zetsuserver.dyndns.org>

<http://it.linkedin.com/in/marcoimpallaria>