

50- batticuore – pulse sensor – cardiofrequenzimetro (some notes at section end)



Il sensore di pulsazione e' uno strumento specifico per la misurazione dei battiti cardiaci. E' composto essenzialmente da un led che illumina il polpastrello di un dito e da un fotoresistore, che ne misura la luce riflessa. Al momento del battito aumenta la presenza di sangue nelle vene e cio' modifica la quantita' di luce riflessa. Questa variazione, catturata dal sensore ed inviata ad Arduino sottoforma di segnale analogico,

consente di contare i batti e quindi calcolare il numero di battiti al secondo.

Il cardiofrequenzimetro presentato in questo esercizio non si limita a contare i battiti, ma e' dotato anche di una routine di "taratura" che analizza le pulsazioni per una decina di secondi, individua i valori massimi e minimi trasmessi dal sensore e definisce la soglia oltre la quale il valore trasmesso dal sensore puo' essere considerato un battito.

La taratura e' necessaria poiche' una soglia fissa e preimpostata non sarebbe in grado di compiere una credibile misurazione per via delle differenze tra le dita e per il fatto che una eccessiva o comunque non ottimale pressione del sensore sul polpastrello incide sulla "portata" delle vene, sia nel momento del battito che nel momento di riposo.

Per ottenere risultati ottimali e' opportuno isolare, con del nastro adesivo trasparente, le superfici del sensore in modo che non entrino in contatto con la pelle, la cui umidita' o il cui sottile strato di olio, potrebbe recare danni. E' anche opportuno assicurare il sensore al polpastrello mediante una striscia di velcro, cosi' come appare nella foto piu' avanti o nel filmato, visibile [qui](#).

Prima di procedere alla compilazione del programma devono essere installate, se non gia' presenti, le seguenti librerie:

- LiquidCrystal_I2C.h reperibile in <https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads>

Per installare la libreria e' necessario:

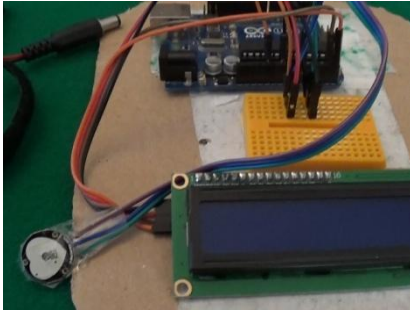
- Fare il download della libreria in formato compresso
- installare la nuova libreria andando in IDE-> sketch-> includes Library-> add .zip library
- verificare l'avvenuta installazione (andando in IDE-> sketch-> includes Library-> Contributed library)

Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Here some notes about this project, translated by google translator



The pulse sensor is a specific tool to measure heart rate. It consists essentially in a LED that illuminates a fingertip, and in a photoresistor which measures the light reflected by finger. It is based on the principle of increased blood volume in the finger veins at pulsation moment, and hence in a slight variation of the reflected light from skin.

This variation, captured by the sensor and sent to Arduino as an analogue signal, allows you to count beats and then, with a certain approximation, to evaluate number of beats per second.

The heart rate monitor presented in this project not only counts beats but also has a "calibration" routine that analyzes pulses for a dozen seconds, detects the maximum and minimum values transmitted by sensor and defines the threshold beyond which the value transmitted can be considered a pulse.

Calibration is necessary since a fixed and predefined threshold would not be able to make a credible measurement due to fistic differences between fingers and because an excessive or a not optimal pressure of sensor on fingertip affects veins both at beat moment and at rest moment.

In order to obtain better results, the sensor surfaces should be insulated with transparent adhesive tape to avoid contact with skin, whose moisture or whose thin layer of oil may damage it. It is also advisable to secure sensor using a velcro strip, as it appears in the movies (found [here](#)) or in the above photo

Before proceeding to program compilation must be installed, if not already done, the library:

- LiquidCrystal_I2C.h found [here](#)

For library installation:

- library download in compressed form;
- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation please verify the library. It must be present in IDE-> sketch-> includes Library-> Contributed library

Note: This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to giocarduino@libero.it (simple words and short sentences, please)

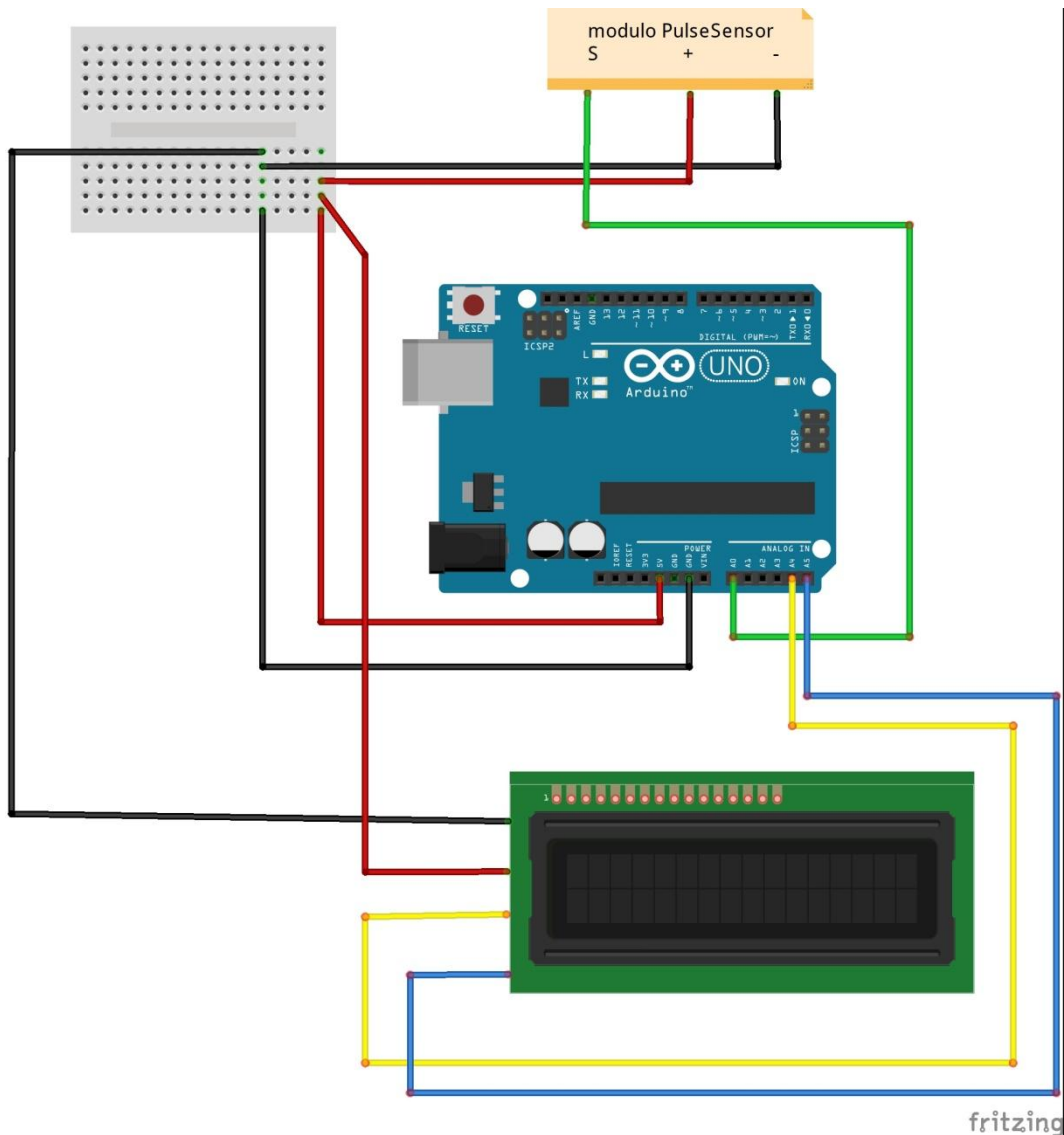
Materiali

- Un sensore di pulsazioni (vedi le figure iniziali) commercializzato da PulseSensor.com e facilmente reperibile su amazon o sui mercati asiatici
- Una striscia di velcro

Arduino: batticuore – pulse sensor – cardiofrequenzimetro

- Un display lcd 1602 con circuito I2C
- Una breadboard
- Un po' di cavetteria

Schema



Programma

/*

Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo. Per rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il trasferimento nell'IDE, premendo CTRL+T.

In questo progetto si utilizza il modulo pubblicizzato in PulseSensor.com per rilevare il battito cardiaco. L'aggeggio e' composto da una sorgente di luce da una fotoresistenza che misura la luce riflessa dal dito. L'idea su cui si e' che nel momento della pulsazione la portata delle vene e' alta e rende quindi piu' problematico il passaggio della luce. Il programma conta quindi i picchi registrati dalla fotoresistenza e, in funzione del tempo trascorso, determina il numero di battiti al minuto. Nel programma e' presente una routine di taratura per il calcolo del valore oltre il quale il segnale analogico inviato dal sensore puo' essere considerato un picco.

La taratura appare necessaria poiche' il valore di confronto varia al variare delle persone e della pressione esercitata dal sensore sul dito.

Arduino: batticuore – pulse sensor – cardiofrequenzimetro

Warning: cut&paste from PDF to IDE loses formatting. To restore it press CTRL + T.

In this project you use the appliance solded by PulseSensor.com to detect the heartbeat. The system is composed by a light source and a photoresistor that measures light reflected by a finger. The idea on which is based is that at pulsation moment the extent of the veins is high and makes more problematic the light passage. The program therefore counts peaks recorded by photoresistor and, depending on the elapsed time, determines the number of beats per minute. There is also a calibration routine to calculate the threshold to consider a peak a signal sent by the sensor. Calibration appears necessary because the comparison value varies at varyng people and at varyng pressure exerted by sensor on finger.

```
*/

# define sensore 0      // il sensore di battito e' collegato alla porta analogica 0
# define traccia 0     // 1 = espone la traccia sul monitor seriale, 0 = non espone la traccia

#include <LiquidCrystal_I2C.h> // libreria di gestione del display lcd
//. . . . . addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // definisce tipologia del display
byte cuoregrande[8] = // la configurazazione dei bit "disegna" il carattere da esporre
{
  0b00000,
  0b01010,
  0b11111,
  0b11111,
  0b11111,
  0b01110,
  0b00100,
  0b00000
};
byte cuorepiccolo[8] =
{
  0b00000,
  0b00000,
  0b01010,
  0b01110,
  0b01110,
  0b00100,
  0b00000,
  0b00000
};

int statosensore           = 0; // zona di memorizzazione dello stato sensore
int semaforo              = 0; // 0 = nessuna misurazione in corso; 1 == misurazione in corso
int contatore             = 0; // contatore di battiti
unsigned long tempoinizio  = 0; // memorizza il momento di inizio rilevamento pulsazioni
unsigned long tempoattuale = 0; // zona di memorizzazione del momento corrente
float tempotrascorso      = 0; // tempo trascorso (in millisecondi) da inizio misurazione
int battitoincorso       = 0; // semaforo utilizzato per evitare di contare piu' volte il
// medesimo battito)
int minimo                = 1000; // valore minimo in fase di taratura
int massimo               = 0; // valore massimo in fase di taratura
int indice                = 0; // indice utilizzato per la taratura
int valoreconfronto       = 0; // valore oltre il quale si considera il battito
int frequenza             = 0; // battiti al minuto
//
// ***** routine di conclusione della misurazione corrente e di ritaratura
// ***** end measuring routine *****
//
void finemisura ()
{
  lcd.clear ();
  lcd.print ("fine misura");
  lcd.setCursor (0, 1);
  lcd.print ("puls/min: ");
  lcd.setCursor (12, 1);
  lcd.print (frequenza);
  lcd.print (" ");
  delay (3000);
  lcd.clear();
  lcd.print ("ritaratura...");
  contatore = 0;
  semaforo = 0;
  battitoincorso = 0;
  taratura ();
}
//
// ***** routine di taratura (calcolo della soglia di confronto)
// *****tare routine (compute threshold) *****
```

Arduino: batticuore – pulse sensor – cardiofrequenzimetro

```
//
void taratura ()
{
  if (traccia)
  {
    Serial.println ("taratura");
  }
  minimo = 1000;
  massimo = 0;
  for (indice = 0; indice < 50; indice++)
  {
    statosensore = analogRead (sensore);
    if (statosensore < minimo)
      minimo = statosensore;
    if (statosensore > massimo)
      massimo = statosensore;
    delay (150);
  }
  if (traccia)
    Serial.println ("fine taratura");
  valoreconfronto = (minimo + ((massimo - minimo) * 0.65));
  if (traccia)
  {
    Serial.print ("valore di confronto ");
    Serial.println (valoreconfronto);
    delay (2000);
  }
  tempoinizio = millis ();
  tempotrascorso = 10;
}
//
//
void setup()
{
  Serial.begin (9600);
  lcd.begin(16, 2);
  lcd.backlight();
  lcd.createChar(1, cuoregrande); // assegna al carattere speciale 1 la configurazione di
// "cuoregrande" *** assign to the special character 1 the "big heart" configuration
  lcd.createChar(2, cuorepiccolo); // assegna al carattere speciale 2 la configurazione di
// "cuorepiccolo" *** assign to the special character 1 the "little heart" configuration
  lcd.clear ();
  lcd.print ("cardiofrequenza ");
  lcd.setCursor (0, 1);
  lcd.print ("sistema pronto");
  delay (1500);
  lcd.clear();
  lcd.print ("taratura...");
  taratura ();
}

void loop()
{
  tempoattuale = millis (); // rileva il momento corrente
  tempotrascorso = (tempoattuale - tempoinizio);
  if ((tempotrascorso > 30000) && (semaforo == 1)) // se sono passati 30 secondi
    finemisura (); // lancia routine di termine misurazione
  statosensore = analogRead (sensore);
  if ((statosensore > valoreconfronto) && ( battitoincorso == 0)) // se c'e' stato un nuovo battito
  {
    battitoincorso = 1; // segnala che e' in atto la misurazione del battito (onde evitare
// di contare due volte il medesimo battito)
    if (semaforo == 0) // se e' il primo battito di una misurazione
    {
      semaforo = 1; // segnala misurazione in corso
      lcd.clear ();
      lcd.print ("misura in corso ");
      contatore = 0; // inizializza il contatore di battiti
      tempoinizio = millis ();
    }
    lcd.setCursor (0, 1);
    lcd.write (2); // espone il cuore piccolo
    contatore++;
    if (contatore > 4) // se sono sati computati almeno quattro battiti, inizia ad esporre
// i risultati
    {
      lcd.setCursor (2, 1);
      lcd.print ("puls/min: ");
    }
  }
}
```

Arduino: batticuore – pulse sensor – cardiofrequenzimetro

```
    lcd.setCursor (12, 1);
    frequenza = ((60000 / tempotrascorso) * contatore);
    lcd.print (frequenza);
    lcd.print (" ");
    if (traccia)
    {
        Serial.print ("battiti al minuto ");
        Serial.println (frequenza);
    }
    delay (300); // attende tre decimi di secondo, per smorzare la pulsazione appena computata
}
else
{
    if (semaforo == 1)
    {
        battitoincorso = 0; // predispone il sistema ad accettare un nuovo battito
        lcd.setCursor (0, 1);
        lcd.write (1); // espone il cuore grande
        delay (50);
    }
}
}
```