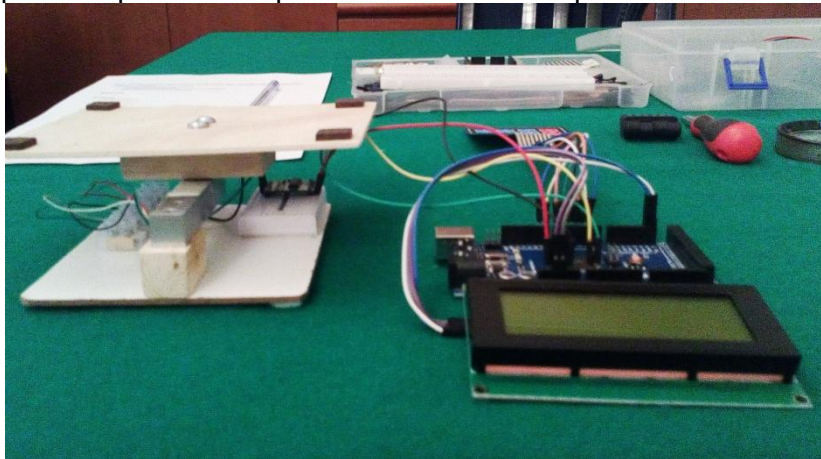
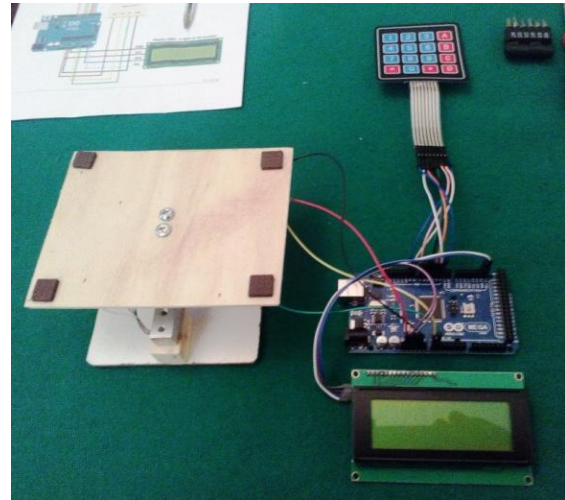


## N - il pesa calorie - the caloriescale (some notes at end of this section)

Una bilancia che oltre ad esporre il peso, calcola anche il numero di kilocalorie del prodotto che ha sul piatto non e' una novita', ma e' comunque un interessante utilizzo di Arduino che in questo progetto oltre a gestire tre differenti device (il display, la tastiera ed il sensore di peso), combina le informazioni fornite dalla tastiera e dal sensore di peso per calcolare, attraverso una tabella interna, il numero di calorie del prodotto sul piatto e quelle dell'intera ricetta in via di preparazione, per poi esporle su di un display da 4 righe di 20 caratteri. Dal punto di vista tecnico sono state unite le esperienze descritte negli esercizi [41 – sensore di peso](#), bilancia digitale e [42 – tastiera numerica e display 2004](#), ai quali si rimanda per le implicazioni tecnico costruttive. In particolare nell'esercizio 41, sensore di peso, sono presenti le indicazioni di taratura della bilancia, indispensabili anche per la costruzione del pesacalorie. Sotto l'aspetto pratico il pesacalorie pesa cio' che trova sul piatto e si mette in attesa di un input da tastiera. Il



primo numero digitato identifica la categoria alimentare mentre il secondo numero identifica l'alimento all'interno della categoria (vedi piu' sotto la tabella delle categorie alimentari e degli alimenti). Il programma calcola il numero di calorie per la quantita' di alimento presente sul piatto, lo espone sul display e, se viene premuto il tasto "b", lo aggiunge al totale della ricetta. In caso di errore di digitazione e' sufficiente

premere il tasto "a" per cancellare la scelta appena effettuata.

Piu' in dettaglio, queste le funzioni dei tasti non numerici

- "a" - cancella la scelta effettuata e mette il sistema in attesa di una nuova scelta ripartendo dalla categoria alimentare
- "b" - aggiunge l'ingrediente, ancora sul piatto, al totale delle calorie della ricetta in via di composizione
- "c" – elimina l'ultimo ingrediente aggiunto alla ricetta
- "d" – cancella tutto e predispone la bilancia ad acquisire una nuova ricetta
- "\*" - conferma eliminazione ultimo ingrediente
- "#" – annulla richiesta di eliminazione

Nel momento in cui la bilancia registra il valore zero (quando si accorge che non c'e' nulla sul piatto) cancella la descrizione sul display e si pone in attesa di un nuovo ingrediente per la ricetta in via di composizione. [Qui il film del progetto.](#)

**Attenzione:** la dimensione della tabella dei prodotti e' tale da renderne proibitivo l'utilizzo con Arduino uno. Questo progetto puo' essere replicato solo se si dispone di una scheda di maggior potenza, tipo **arduino mega 2560**, che dispone di 256k di memoria al posto dei 32k di Arduino uno.

**Tabella alimenti e calorie per 100 grammi**

cat. alimentare 0 - verdura ed ortaggi freschi

| codice | descrizione                 | Kcal |
|--------|-----------------------------|------|
| 0      | carote e porri              | 32   |
| 1      | cipolle e cavoli            | 25   |
| 2      | fagiolini e verze           | 20   |
| 3      | funghi                      | 23   |
| 4      | melanzane, carciofi, zucca  | 18   |
| 5      | patate e legumi             | 80   |
| 6      | peperoni                    | 22   |
| 7      | pomodori                    | 19   |
| 8      | insalata, finocchi e sedano | 15   |
| 9      | zucchine e cetrioli         | 14   |

cat. alimentare 1 - frutta (fresca e secca)

| codice | descrizione                | Kcal |
|--------|----------------------------|------|
| 0      | albicocche, pesche, meloni | 28   |
| 1      | arance                     | 34   |
| 2      | castagne e prugne secche   | 180  |
| 3      | fichi e prugne fresche     | 50   |
| 4      | fichi secchi               | 246  |
| 5      | mandorle e pinoli          | 575  |
| 6      | mele e pere                | 43   |
| 7      | nocci, nocciole e arachidi | 630  |
| 8      | uva e banana               | 63   |

cat. alimentare 2 - carni fresche e uova

| codice | descrizione          | Kcal |
|--------|----------------------|------|
| 0      | agnello              | 162  |
| 1      | anatra               | 159  |
| 2      | cappone              | 226  |
| 3      | coniglio e cinghiale | 120  |
| 4      | maiale               | 180  |
| 5      | manzo                | 160  |
| 6      | pollo e tacchino     | 130  |
| 7      | uova intere          | 130  |
| 8      | uova tuorlo          | 325  |
| 9      |                      |      |

cat. alimentare 3 - pesce fresco e conservato

| codice | descrizione                | Kcal |
|--------|----------------------------|------|
| 0      | Alici e acciughe salate    | 128  |
| 1      | alici e acciughe fresche   | 110  |
| 2      | Branzino e merluzzo        | 82   |
| 3      | crostacei                  | 85   |
| 4      | seppie, polpi e calamari   | 70   |
| 5      | salmone affumicato         | 180  |
| 6      | sardine e orate fresche    | 120  |
| 7      | sardine e tonno con olio   | 250  |
| 8      | sgombro, tonno, salmone fr | 160  |
| 9      | trota fresca               | 100  |

cat. alimentare 4 - latte e latticini

| codice | descrizione               | Kcal |
|--------|---------------------------|------|
| 0      | formaggi freschi          | 240  |
| 1      | formaggi stagionati duri  | 400  |
| 2      | formaggi stagionati molli | 310  |
| 3      | latte e yogurt intero     | 64   |
| 4      | latte e yogurt scremato   | 46   |
| 5      | mascarpone                | 460  |
| 6      | panna                     | 300  |
| 7      | ricotta                   | 146  |
| 8      |                           |      |
| 9      |                           |      |

cat. alimentare 5 - carni conservate

| codice | descrizione      | Kcal |
|--------|------------------|------|
| 0      | prosciutto crudo | 159  |
| 1      | prosciutto cotto | 138  |
| 2      | speck            | 303  |
| 3      | mortadella       | 317  |
| 4      | Salsiccia fresca | 170  |
| 5      | salame           | 400  |
| 6      | pancetta         | 661  |
| 7      | bresaola         | 153  |
| 8      |                  |      |
| 9      |                  |      |

cat. alimentare 6 - cereali e derivati

| codice | descrizione              | Kcal |
|--------|--------------------------|------|
| 0      | frollini e bisc. di riso | 465  |
| 1      | farine e fecole          | 350  |
| 2      | Fette e biscotti secchi  | 390  |
| 3      | gnocchi di patate        | 156  |
| 4      | pane condito             | 285  |
| 5      | pane e pasta fresca      | 270  |
| 6      | pane integrale           | 230  |
| 7      | pasta integrale secca    | 335  |
| 8      | pasta ripiena fresca     | 280  |
| 9      | riso e pasta secca       | 362  |

cat. alimentare 7 - condimenti e prodotti semilav.

| codice | descrizione              | Kcal |
|--------|--------------------------|------|
| 0      | olio e strutto           | 900  |
| 1      | burro                    | 750  |
| 2      | dado per brodo           | 180  |
| 3      | lardo                    | 891  |
| 4      | margarina                | 760  |
| 5      | concentrato di pomodoro  | 50   |
| 6      | pelati e passata di pom. | 20   |
| 7      | legumi secchi            | 310  |
| 8      | orzo perlato             | 319  |
| 9      |                          |      |

cat. alimentare 8 - prodotti industriali

| codice | descrizione            | Kcal |
|--------|------------------------|------|
| 0      | fiocchi di cereali     | 360  |
| 1      | frutta candita         | 250  |
| 2      | fagioli e mais scatola | 90   |
| 3      | marmellata             | 284  |
| 4      | miele                  | 204  |
| 5      | olive in salamoia      | 110  |
| 6      | piselli in scatola     | 55   |
| 7      | semolino               | 350  |
| 8      | zucchero               | 392  |
| 9      | nutella e cioccolato   | 533  |

cat. alimentare 9 - bevande

| codice | descrizione               | Kcal |
|--------|---------------------------|------|
| 0      | bibite gassate zuccherate | 45   |
| 1      | bibite light              | 22   |
| 2      | birra                     | 42   |
| 3      | liquore 40 gradi          | 240  |
| 4      | vino 12 gradi             | 66   |
| 5      |                           |      |
| 6      |                           |      |
| 7      |                           |      |
| 8      |                           |      |
| 9      |                           |      |

## Arduino: il pesa calorie – the caloriescale

Prima di compilare il programma bisogna installare, se non già fatto, le librerie:

- LiquidCrystal\_I2C.h reperibile [qui](#)
- HX771.h reperibile [qui](#)

Per installare le librerie:

- Scaricare le librerie in formato compresso;
- Installarle via IDE-> sketch-> includes Library-> add .zip library
- Dopo l'installazione verificarne la presenza in: IDE-> sketch-> includes Library-> Contributed library

Deve anche essere installata la libreria "keypad.h" library, presente nel library manager dell'IDE (sketch->include library->manages libraries), ma non ancora installata

**Nota:** Questo esercizio e questa scheda 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](mailto:giocarduino@libero.it)

### Here some notes about this project, translated by google translator



A scale that in addition to exhibiting weight, also calculates the product kilocalories is not something new, but is an interesting use of Arduino. In this project, Arduino runs three different devices (display, keyboard and load cell) and combines the information from keyboard and from load cell to calculate, through an internal array, calories of product on plate and those recipe being prepared.

From a technical standpoint were combined the experiences described in projects: 41 - weight sensor and 42 - numeric keypad and display 2004, which must be read for the constructive technical implications. In particular, in project 41, weight sensor, are detailed the directions for digital scale calibration, **also needed in construction of this appliance.**

Under the practical aspect, the caloriescale weighs what is located on plate and waits for an input from keyboard. First dialed number identifies the food category and the second number identifies the food inside of the category (See under the list of food categories and food). The program calculates number of calories for the amount food on plate, shows it on lcd display and, if you press the "b" button, adds it to total of recipe. In case of typing error, simply press the "a" key to delete the selection you just made

Specifically, these are the functions of the non-numeric keys

- "A" - erases your choice and puts the system waiting for a new choice, starting from food category
- "B" - adds the ingredient, still on the plate, to recipe total calories
- "C" - deletes the last ingredient added to recipe
- "D" - deletes everything and prepares the scale to acquire a new recipe
- "\*" - Confirms elimination last ingredient
- "#" - Cancel delete request

At a time when the scale registers the value zero (when it realizes that nothing is on plate), clears any description on screen and waits for a new ingredient for recipe in progress. [Here the movie.](#)

## Arduino: il pesa calorie – the caloriescale

**Warning:** the size of food table is prohibitive for Arduino uno. This project can be replicated only if you have a more powerful board, type **Arduino Mega 2560**, which has 256k of memory instead of the 32k of Arduino uno.

### list foods and calories per 100 grams

food category 0 - fresh vegetables

| code | description                   | Kcal |
|------|-------------------------------|------|
| 0    | carrots or leeks              | 32   |
| 1    | onions or cabbage             | 25   |
| 2    | green (beans or cabbage)      | 20   |
| 3    | mushrooms                     | 23   |
| 4    | eggplant, artichokes, pumpkin | 18   |
| 5    | Potatoes or legumes           | 80   |
| 6    | pepperoni                     | 22   |
| 7    | tomatoes                      | 19   |
| 8    | salad, fennel and celery      | 15   |
| 9    | Zucchini and cucumbers        | 14   |

food category 1 - fruit (fresh or dried)

| code | description                 | Kcal |
|------|-----------------------------|------|
| 0    | apricots, peaches, melon    | 28   |
| 1    | oranges                     | 34   |
| 2    | chestnuts and dry prunes    | 180  |
| 3    | figs and fresh plums        | 50   |
| 4    | Dry figs                    | 246  |
| 5    | almonds and pine nuts       | 575  |
| 6    | apples and pears            | 43   |
| 7    | walnuts, hazelnuts, peanuts | 630  |
| 8    | Grapes and bananas          | 63   |
| 9    |                             |      |

food category 2 - meat and fresh eggs

| code | description          | Kcal |
|------|----------------------|------|
| 0    | lamb                 | 162  |
| 1    | duck                 | 159  |
| 2    | capon                | 226  |
| 3    | rabbit and wild boar | 120  |
| 4    | pig                  | 180  |
| 5    | beef                 | 160  |
| 6    | chicken and turkey   | 130  |
| 7    | whole eggs           | 130  |
| 8    | egg yolk             | 325  |
| 9    |                      |      |

food category 3 - fresh fish and preserved

| code | description                   | Kcal |
|------|-------------------------------|------|
| 0    | salted anchovies              | 128  |
| 1    | Fresh anchovies               | 110  |
| 2    | sea bass and cod              | 82   |
| 3    | shellfish                     | 85   |
| 4    | cuttlefish, octopus and squid | 70   |
| 5    | smoked salmon                 | 180  |
| 6    | Fresh sardines and sea bream  | 120  |
| 7    | sardines and tuna with oil    | 250  |
| 8    | Fresh mackerel, tuna, salmon  | 160  |
| 9    | fresh trout                   | 100  |

food category 4 - milk and dairy products

| code | description             | Kcal |
|------|-------------------------|------|
| 0    | Fresh cheese            | 240  |
| 1    | hard aged cheeses       | 400  |
| 2    | Soft ripened cheeses    | 310  |
| 3    | Whole milk and yogurt   | 64   |
| 4    | Skimmed milk and yogurt | 46   |
| 5    | mascarpone cheese       | 460  |
| 6    | cream                   | 300  |
| 7    | ricotta cheese          | 146  |
| 8    |                         |      |
| 9    |                         |      |

food category 5 - carni conservate

| code | description   | Kcal |
|------|---------------|------|
| 0    | ham           | 159  |
| 1    | baked ham     | 138  |
| 2    | speck         | 303  |
| 3    | mortadella    | 317  |
| 4    | fresh sausage | 170  |
| 5    | salami        | 400  |
| 6    | beacon        | 661  |
| 7    | bresaola      | 153  |
| 8    |               |      |
| 9    |               |      |

food category 6 - cereals

| code | description                  | Kcal |
|------|------------------------------|------|
| 0    | shortbread and rice biscuits | 465  |
| 1    | flours and starches          | 350  |
| 2    | dry biscuits and rusks       | 390  |
| 3    | gnocchi                      | 156  |
| 4    | Seasoned bread               | 285  |
| 5    | Fresh bread and pasta        | 270  |
| 6    | wholemeal bread              | 230  |
| 7    | dry wholemeal pasta          | 335  |
| 8    | fresh stuffed pasta          | 280  |
| 9    | rice and dry pasta           | 362  |

food category 7 - condiments and processed prdct

| code | description               | Kcal |
|------|---------------------------|------|
| 0    | oil and melted lard       | 900  |
| 1    | butter                    | 750  |
| 2    | stock cube                | 180  |
| 3    | lard                      | 891  |
| 4    | margarine                 | 760  |
| 5    | tomato concentrate        | 50   |
| 6    | tomatoes and tomato puree | 20   |
| 7    | dried vegetables          | 310  |
| 8    | Pearl barley              | 319  |
| 9    |                           |      |

food category 8 - industrial products

| code | description           | Kcal |
|------|-----------------------|------|
| 0    | cereal flakes         | 360  |
| 1    | candied fruit         | 250  |
| 2    | Canned beans and mais | 90   |
| 3    | jam                   | 284  |
| 4    | honey                 | 204  |
| 5    | olives in brine       | 110  |
| 6    | canned peas           | 55   |
| 7    | semolina              | 350  |
| 8    | sugar                 | 392  |
| 9    | nutella and chocolate | 533  |

food category 9 - drinks

| code | description             | Kcal |
|------|-------------------------|------|
| 0    | fizzy drinks and sugary | 45   |
| 1    | Light drinks            | 22   |
| 2    | beer                    | 42   |
| 3    | liquor 40 degrees       | 240  |
| 4    | vine 12 degrees         | 66   |
| 5    |                         |      |
| 6    |                         |      |
| 7    |                         |      |
| 8    |                         |      |
| 9    |                         |      |

## Arduino: il pesa calorie – the caloriescale

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

- LiquidCrystal\_I2C.h found [here](#)
- HX771.h found [here](#)

For library installation, see process shown in previous projects, and summarized in:

- 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

Must also be installed the “keypad.h” library, present but non yet installed in the library manager (sketch->include library->manages libraries)

**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.

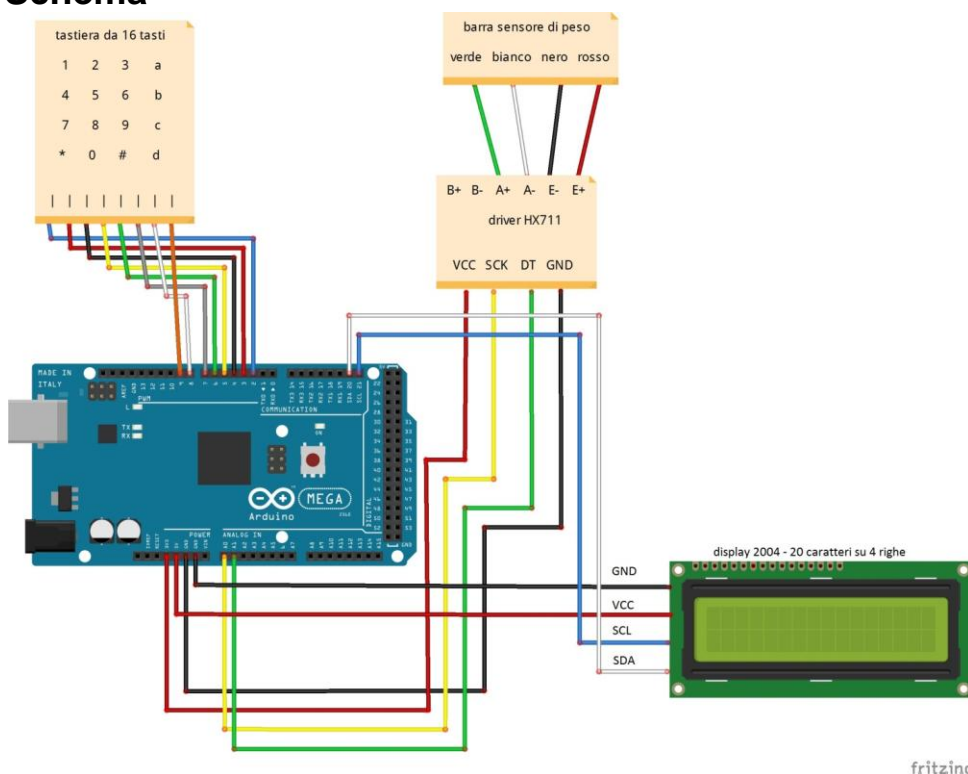
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

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

## Materiali

- Un Arduino mega 2560
- Un sensore di peso e relativo driver HX711
- Una tastiera da 16 tasti
- Un display 2004 (20 caratteri su quattro righe) con driver I2C
- Un po' di cavetti
- Un pacco batterie da 9 volt

## 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. Nel trasferimento si perdono anche gli spazi compresi
 * tra apici (" "), per cui una volta trasferito il programma nell'IDE bisogna intervenire
 * manualmente per reintrodurre gli spazi perduti (vedi anche le note in rosso nella versione pdf
 * del programma).
 *
 * Questo programma contiene, oltre al set di caratteri utilizzati dalla libreria liquidcrystal,
 * una tabella di oltre 2200 caratteri e per tale ragione non puo' essere eseguito su Arduino uno
 * (che dispone, per le variabili, poco piu' di 2000 caratteri). E' stato quindi sviluppato e
 * provato su Arduino mega che, per caratteristiche e potenza, puo' tranquillamente gestire
 * tabelle di grandi dimensioni.
 *
 * Questo programma unisce le esperienze degli esercizi 41 e 42 (sensore di peso e
 * tastiera + display2004), alle cui schede si rimanda per la costruzione
 * dell'impianto e per le problematiche di installazione delle librerie (attenzione:
 * il display 2004 deve essere collegato alle porte 20 e 21 di arduino mega al posto
 * delle tradizionali porte A4 ed A5 di arduino uno).
 *
 * connessioni del modulo HX711:
 * Hx711.DO - porta analogica 1
 * Hx711.SCK - porta analogica 0
 *
 * connessioni del sensore di peso:
 * rosso = E+
 * nero = E-
 * bianco = A-
 * verde = A+
 *
 * connessioni del tastierino numerico:
 * pin delle righe (primi 4 connettori di sinistra): porte 2,3,4 e 5
 * pin delle colonne (ultimi 4 connettori di destra): porte 6,7,8,e 9
 *
 * connessione del display 2004 (4 righe da 20 caratteri)
 * pin SDA: porta 20
 * pin SCL: porta 21
 *
 * Manuale d'uso:
 * tasti numerici: selezione della categoria alimentare e dell'alimento
 * tasti non numerici:
 * "A" = clear, cancella l'ultima selezione e chiede nuovamente la categoria alimentare
 * per il prodotto presente sul piatto
 * "B" = aggiunge alla ricetta l'ingrediente sul piatto
 * "C" = eliminazione dalla ricetta l'ultimo ingrediente aggiunto
 * "D" = pulisce il sistema e lo predispone ad una nuova ricetta
 * "*" = conferma l'eliminazione
 * "#" = annulla la richiesta di eliminazione
 *
 * -----
 * Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T. In
 * transfer will also loses the spaces in quotes (""), so once transferred program in IDE,
 * we must intervene manually to introduce the lost spaces (see also red notes on pdf version
 * of program).
 *
 * This program contains, in addition to the character set used by liquid crystal library, an array
 * of more than 2200 characters and for this reason can not runs on Arduino uno (which has, for
 * variables, a little more than 2000 characters). It was developed and tested on Arduino Mega that,
 * for features and power, can safely handle a large array.
 *
 * HX711: device connections :
 * Hx711.DO - analog pin 1
 * Hx711.SCK - analog pin 0
 *
 * load cell connections to hx711
 * red = E+
 * balck = E-
 * withe = A-
 * green = A+
 *
 * numeric keypad connections:
 * rows pins (firs 4 connector on the left): pins 2,3,4 and 5
 * columns pins (last 4 connectors on the rigth): pins 6,7,8,and 9
 *
 * display 2004 connections:
 * SDA: pin 20
```

## Arduino: il pesa calorie – the caloriescale

```
* SCL: pin 21
*
* user manual:
* numeric keys: select category and food
* non numeric keys:
* "A" = clear last selection and restart from food category for product on plate
* "B" = adds to recipe the on plate food
* "C" = delete, from recipe, the last food added
* "D" = clear system and start with a new recipe
* "*" = confirm delete
* "#" = cancels the deletion request *
*
*/
#include <Keypad.h>           // libreria di gestione della tastiera
#include <Wire.h>             // libreria wire presente, di fault, nell'IDE
#include <LiquidCrystal_I2C.h> // libreria di gestione del display lcd
#include "HX711.h"           // libreria di gestione del modulo HX711, per la gestione
//                             del sensore di peso
//
// nelle prossime due righe vengono attribuiti i parametri per la gestione del display
//-----addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

#define DOUT A1              // porta cui deve essere connesso il pin DO del driver HX711
#define CLK A0               // porta cui deve essere connesso il pin CLK del driver HX711
HX711 bilancia(DOUT, CLK);
int      peso                = 0; // zona di memorizzazione del peso corrente
int      pesoprec            = 0; // zona di memorizzazione dell'ultimo peso esposto
float    zonacalcolo        = 0; // zona di transito per i calcoli
char     tastobattuto;     // variabile di memorizzazione del carattere battuto
const byte righe            = 4; // numero di righe della tastiera (quattro)
const byte colonne          = 4; // numero di colonne della tastiera (quattro)
char     chiave[righe][colonne] = {
// tabella di trascodifica dei tasti battuti.
// la tabella riproduce, su quattro righe di quattro colonne, la posizione dei tasti
// e assegna ad ogni tasto il valore che gli compete. E'possibile assegnare valori
// diversi semplicemente modificando i singoli valori in tabella
//-----
// transcoding table of keystrokes. The table reproduces, on four rows of four columns, the
// location of the keys and assigns to each key a value. You can assign different values simply by
// changing the individual values in the table
//-----
//
  {'1', '2', '3', 'a'},
  {'4', '5', '6', 'b'},
  {'7', '8', '9', 'c'},
  {'*', '0', '#', 'd'}
};
byte pinrighe[righe]      = {2, 3, 4, 5}; //porte cui devono essere connessi i pin delle
// righe (i primi quattro pin di sinistra) della tastiera - rows pins (on connector): first four
// pins on the left)
//
byte pincolonne[colonne] = {6, 7, 8, 9}; //porte cui devono essere connessi i pin delle
// colonne (gli ultimi quattro pin di destra) della tastiera - columns pins (on connector): last
// four pin on the right)
//
//prossima riga: assegnazione dei parametri alla libreria keypad.h - next line: parameters to
// keypad.h library
Keypad keypad = Keypad( makeKeymap(chiave), pinrighe, pincolonne, righe, colonne );
//
int      tastocorrente    = 0; // zona di memorizzazione dell'ultimo tasto numerico battuto
unsigned long iniziopesostabile = 0; // zona di memorizzazione del momento di inizio del controllo
// della stabilizzazione del peso
unsigned long calalim      = 0; // zona di calcolo intermedio
unsigned long comodocalcolo = 0; // zona di calcolo intermedio
int semaforo              = 0; /* semaforo che indica la situazione del sistema:
0 = si e' in attesa della battitura del primo tasto
1 = si e' in attesa della battitura del secondo tasto
2 = si e' in attesa della battitura di un codice di conferma o annulla
3 = si e' in attesa di battitura del tasto "b" (aggiungi alla ricetta) o del tasto "a" (pulisci)
6 = e' in corso l'azzeramento della ricetta
*
semaphore that indicates the situation of the system:
0 = is waiting for the first key typing
1 = is waiting for the second key
2 = is waiting for a confirmation code or cancels
3 = is waiting for "b" (add to the recipe) or "a" key (clear)
6 = is erasing a recipe
```

## Arduino: il pesa calorie – the caloriescale

```
*/
int pesostabile = 0; // semaforo di segnalazione di peso stabile:
// 0 = peso in via di definizione
// 1 = peso stabile, sistema in fase di calcolo calorie
int catalim = 0; // zona di memorizzazione del codice della categoria alimentare
int alim = 0; // zona di memorizzazione del codice dell'alimento
int cal = 0; // zona di memorizzazione delle calorie dell'alimento che si trova sul
piatto
int totcal = 0; // zona di memorizzazione delle calorie della ricetta in via di
elaborazione
int ultcat = 0; // zona di memorizzazione della categoria dell'ultimo alimento aggiunto
alla ricetta
int ultalim = 0; // zona di memorizzazione del codice dell'ultimo alimento aggiunto alla
ricetta
int ultpeso = 0; // zona di memorizzazione del peso dell'ultimo ingrediente aggiunto alla
ricetta
int ultcal = 0; // zona di memorizzazione delle calorie dell'ultimo ingrediente aggiunto
alla ricetta
//
char categoria [10][21] = // tabella categorie alimentari - food category array
{
  {"verdura,ortaggi frsc"},
  {"frutta fresca/secca "},
  {"carni fresche e uova"},
  {"pesce fresco/conserv"},
  {"latte e latticini "}, // riempire con spazi a destra sino a completare i 20 caratteri
  {"salumi/carni conserv"}, // array to be translated for english version (20 characers each)
  {"pane/cereali e deriv"},
  {"condimenti e semilav"},
  {"prodotti lavorati "}, // riempire con spazi a destra sino a completare i 20 caratteri
  {"bevande "}, // riempire con spazi a destra sino a completare i 20 caratteri
};

// le seguenti tabelle: alimento e calorie sono tabelle gemelle,
// sono cioe' tabelle che, a parita' di indice, forniscono una la descrizione dell'alimento
// e l'altra il numero di calorie per 100 grammi
//
// the array: "alimento" and "calorie" are twin array, they uses same index to provide food
// description and number of calories per 100 grams
//
char alimento [100][21] =
{
  {"carote e porri "}, // inserire, in tutte le 100 righe di questa tabella, spazi a
  {"cipolle o cavoli "}, // destra sino a completare i 20 caratteri all'interno di ogni
  {"fagiolini o verze "}, // indice 2 {" "}
  {"funghi freschi "}, // indice 3 - array to be translated for english versione (20 char each)
  {"melanz/carciof/zucca"}, // indice 4
  {"leguni o patate "}, // indice 5
  {"peperoni "}, // indice 6
  {"pomodori "}, // indice 7
  {"insalata/finoc/sedan"}, // indice 8
  {"zucchine e cetrioli "}, // indice 9
  {"pesche/albic/meloni "}, // indice 10
  {"arance "}, // indice 11
  {"castagne/prugne secc"}, // indice 12
  {"fichi/prugne fresche"}, // indice 13
  {"fichi secchi "}, // indice 14
  {"mandorle e pinoli "}, // indice 15
  {"mele e pere fresche "}, // indice 16
  {"noci nocciole arach "}, // indice 17
  {"uva o banane fresche"}, // indice 18
  {"codice non valido "}, // indice 19
  {"agnello "}, // indice 20
  {"anatra "}, // indice 21
  {"cappone "}, // indice 22
  {"cinghiale o coniglio"}, // indice 23
  {"maiale "}, // indice 24
  {"manzo "}, // indice 25
  {"pollo o tacchino "}, // indice 26
  {"uova intere "}, // indice 27
  {"tuorlo d'uovo "}, // indice 28
  {"codice non valido "}, // indice 29
  {"acciughe salate "}, // indice 30
  {"acciughe/alici fresc"}, // indice 31
  {"branzino o merluzzo "}, // indice 32
  {"crostacei "}, // indice 33
  {"molluschi "}, // indice 34
  {"salmone affumicato "}, // indice 35
```



## Arduino: il pesa calorie – the caloriescale

```
{ "sardine o orate      " }, // indice 36
{ "sardine tonno olio " }, // indice 37
{ "sgombro tonno fresco" }, // indice 38
{ "trota fresca       " }, // indice 39
{ "formaggio fresco   " }, // indice 40
{ "formaggio stag. duro" }, // indice 41
{ "formaggio stag molle" }, // indice 42
{ "latte/yogourt intero" }, // indice 43
{ "latte/yogourt screm." }, // indice 44
{ "mascarpone         " }, // indice 45
{ "panna              " }, // indice 46
{ "ricotta            " }, // indice 47
{ "codice non valido  " }, // indice 48
{ "codice non valido  " }, // indice 49
{ "prosciutto crudo   " }, // indice 50
{ "prosciutto cotto   " }, // indice 51
{ "speck              " }, // indice 52
{ "mortadella         " }, // indice 53
{ "salame             " }, // indice 54
{ "salsiccia fresca  " }, // indice 55
{ "pancetta           " }, // indice 56
{ "bresaola          " }, // indice 57
{ "codice non valido  " }, // indice 58
{ "codice non valido  " }, // indice 59
{ "frollini/bisc. riso" }, // indice 60
{ "farine e fecole    " }, // indice 61
{ "fette/biscotti secch" }, // indice 62
{ "gnocchi di patate  " }, // indice 63
{ "pane condito       " }, // indice 64
{ "pane e pasta fresca" }, // indice 65
{ "pane integrale     " }, // indice 66
{ "pasta integr secca " }, // indice 67
{ "pasta secca/riso   " }, // indice 68
{ "pasta ripiena fresca" }, // indice 69
{ "olio o strutto     " }, // indice 70
{ "burro              " }, // indice 71
{ "dado per brodo     " }, // indice 72
{ "lardo              " }, // indice 73
{ "margarina          " }, // indice 74
{ "concentr di pomodoro" }, // indice 75
{ "pelati/passata pomod" }, // indice 76
{ "legumi secchi      " }, // indice 77
{ "orzo perlato       " }, // indice 78
{ "codice non valido  " }, // indice 79
{ "fiocchi di cereali " }, // indice 80
{ "frutta candita     " }, // indice 81
{ "fagioli/mais scatola" }, // indice 82
{ "marmellata         " }, // indice 83
{ "miele              " }, // indice 84
{ "olive in salamoia  " }, // indice 85
{ "piselli in scatola " }, // indice 86
{ "semolino           " }, // indice 87
{ "zucchero           " }, // indice 88
{ "nutella o cioccolato" }, // indice 89
{ "bibite gassate zucc." }, // indice 90
{ "bibite lighth      " }, // indice 91
{ "birra              " }, // indice 92
{ "liquore 40 gradi   " }, // indice 93
{ "vino 12 gradi      " }, // indice 94
{ "codice non valido  " }, // indice 95
{ "codice non valido  " }, // indice 96
{ "codice non valido  " }, // indice 97
{ "codice non valido  " }, // indice 98
{ "codice non valido  " }, // indice 99
};

int calorie [100] = // calories tab - uses the same index of previus array
{
  32, // indice 00 {"carote e porri      ", 32},
  25, // indice 01 {"cipolle o cavoli    ", 25},
  20, // indice 02 {"fagiolini o verze   ", 20},
  23, // indice 03 {"funghi freschi     ", 23},
  18, // indice 04 {"melanz/carciof/zucca", 18},
  80, // indice 05 {"leguni o patate     ", 80},
  22, // indice 06 {"peperoni            ", 22},
  19, // indice 07 {"pomodori            ", 19},
  15, // indice 08 {"insalata/finoc/sedan", 15},
  14, // indice 09 {"zucchine e cetrioli ", 14},
```

## Arduino: il pesa calorie – the caloriescale

```
28, // indice 10 {"pesche/albic/meloni ", 28},
34, // indice 11 {"arance ", 34},
180, // indice 12 {"castagne/prugne secc", 180},
50, // indice 13 {"fichi/prugne fresche", 50},
246, // indice 14 {"fichi secchi ", 246},
575, // indice 15 {"mandorle e pinoli ", 575},
43, // indice 16 {"mele e pere fresche ", 43},
630, // indice 17 {"noci nocciole arach ", 630},
63, // indice 18 {"uva o banane fresche", 63},
0, // indice 19 {" ", 0},
162, // indice 20 {"agnello ", 162},
159, // indice 21 {"anatra ", 159},
226, // indice 22 {"cappone ", 226},
120, // indice 23 {"cinghiale o coniglio", 120},
180, // indice 24 {"maiale ", 180},
160, // indice 25 {"manzo ", 160},
130, // indice 26 {"pollo o tacchino ", 130},
130, // indice 27 {"uova intere ", 130},
325, // indice 28 {"tuorlo d'uovo ", 325},
0, // indice 29 {" ", 0},
128, // indice 30 {"acciughe salate ", 128},
110, // indice 31 {"acciughe/alici fresc", 110},
82, // indice 32 {"branzino o merluzzo ", 82},
85, // indice 33 {"crostacei ", 85},
70, // indice 34 {"molluschi ", 70},
180, // indice 35 {"salmone affumicato ", 180},
120, // indice 36 {"sardine o orate ", 120},
250, // indice 37 {"sardine tonno olio ", 250},
160, // indice 38 {"sgombro tonno fresco", 160},
100, // indice 39 {"trota fresca ", 100},
240, // indice 40 {"formaggio fresco ", 240},
400, // indice 41 {"formaggio stag. duro", 400},
310, // indice 42 {"formaggio stag molle", 310},
64, // indice 43 {"latte/yogourt intero", 64},
46, // indice 44 {"latte/yogourt screm.", 46},
460, // indice 45 {"mascarpone ", 460},
300, // indice 46 {"panna ", 300},
146, // indice 47 {"ricotta ", 146},
0, // indice 48 {" ", 0},
0, // indice 49 {" ", 0},
159, // indice 50 {"prosciutto crudo ", 159},
138, // indice 51 {"prosciutto cotto ", 138},
303, // indice 52 {"speck ", 303},
317, // indice 53 {"mortadella ", 317},
400, // indice 54 {"salame ", 400},
170, // indice 55 {"salsiccia fresca ", 170},
661, // indice 56 {"pancetta ", 661},
153, // indice 57 {"bresaola ", 153},
0, // indice 58 {" ", 0},
0, // indice 59 {" ", 0},
465, // indice 60 {"frollini/bisc. riso", 465},
350, // indice 61 {"farine e fecole ", 350},
390, // indice 62 {"fette/biscotti secch", 390},
156, // indice 63 {"gnocchi di patate ", 156},
285, // indice 64 {"pane condito ", 285},
270, // indice 65 {"pane e pasta fresca ", 270},
230, // indice 66 {"pane integrale ", 230},
335, // indice 67 {"pasta integr secca ", 335},
362, // indice 68 {"pasta secca/riso ", 362},
280, // indice 69 {"pasta ripiena fresca", 280},
900, // indice 70 {"olio o strutto ", 900},
750, // indice 71 {"burro ", 750},
180, // indice 72 {"dado per brodo ", 180},
891, // indice 73 {"lardo ", 891},
760, // indice 74 {"margarina ", 760},
50, // indice 75 {"concentr di pomodoro", 50},
20, // indice 76 {"pelati/passata pomod", 20},
310, // indice 77 {"legumi secchi ", 310},
319, // indice 78 {"orzo perlato ", 319},
0, // indice 79 {" ", 0},
360, // indice 80 {"fiocchi di cereali ", 360},
250, // indice 81 {"frutta candita ", 250},
90, // indice 82 {"fagioli/mais scatola", 90},
284, // indice 83 {"marmellata ", 284},
204, // indice 84 {"miele ", 204},
110, // indice 85 {"olive in salamoia ", 110},
55, // indice 86 {"piselli in scatola ", 55},
350, // indice 87 {"semolino ", 350},
```

## Arduino: il pesa calorie – the caloriescale

```
392, // indice 88 {"zucchero", 392},
533, // indice 89 {"nutella o cioccolato", 533},
45, // indice 90 {"bibite gassate zucc.", 45},
22, // indice 91 {"bibite lighth", 10},
42, // indice 92 {"birra", 42},
240, // indice 93 {"liquore 40 gradi", 240},
66, // indice 94 {"vino 12 gradi", 66},
0, // indice 95 {"", 0},
0, // indice 96 {"", 0},
0, // indice 97 {"", 0},
0, // indice 98 {"", 0},
0, // indice 98 {"", 0},
};
//
// ***** routine di pesatura *****
// ***** Weighing routine *****
//
void pesatura (void)
{
  while ((millis() - iniziopesostabile) < 2000) // se il peso e' stabile da meno di 2 secondi
    // verifica che non sia cambiato
  {
    peso = bilancia.get_units(20), 3;
    zonacalcolo = pesoprec - peso;
    if (peso < 3) // se sulla bilancia ci sono meno di 3 grammi
    {
      iniziopesostabile = millis (); // non considera il valore e mette il sistema in attesa di
      // un nuovo peso
      lcd.clear (); // pulisce lo schermo
      lcd.setCursor(0, 0); // posiziona il cursore all'inizio della prima riga (car.0 e riga0)
      lcd.print(peso);
      lcd.print("g ");
      lcd.setCursor(0, 1); // posiziona cursore a inizio della seconda riga (car.0 e riga 1)
      lcd.print("metti sul piatto ");
      lcd.setCursor(0, 2); // posiziona cursore a inizio della terza riga (car.0 e riga 1)
      lcd.print("un nuovo ingrediente");
      lcd.setCursor(0, 3); // posiziona cursore a inizio quarta riga (car. 0 e riga 3)
      lcd.print("tot.kcal: ");
      lcd.print (totcal);
    }
    if (((zonacalcolo * 100) / pesoprec) > 2) || (((zonacalcolo * 100) / pesoprec) < -2)) // se
    // c'e' stata un'escursione di peso maggiore del 2%
    {
      iniziopesostabile = millis (); // e' variato il peso, ricomincia da capo
      pesoprec = peso; // memorizza il peso corrente
      lcd.clear (); // pulisce lo schermo
      lcd.setCursor(0, 0); // posiziona cursore a inizio prima riga (carattere 0 e riga 0)
      lcd.print(peso);
      lcd.print("g - attendi "); // inserire 6 spazi a destra, prima dell'apice di chiusura
      lcd.setCursor(0, 1); // posiziona cursore a inizio secondata riga (carattere 0 e riga 1)
      lcd.print("la stabilizzazione");
      lcd.setCursor(0, 2); // posiziona cursore a inizio terza riga (carattere 0 e riga 2)
      lcd.print("della pesatura... ");
      lcd.setCursor(0, 3); // posiziona cursore a inizio quarta riga (carattere 0 e riga 3)
      lcd.print("tot.kcal: ");
      lcd.print (totcal);
    }
  }
  // se il peso e' stabile da piu' di due secondi
  lcd.clear (); // pulisce lo schermo
  lcd.setCursor(0, 0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print(peso);
  lcd.print("g - categ. alim.?");
  lcd.setCursor(0, 3); // posiziona cursore a inizio quarta riga (carattere 0 e riga 3)
  lcd.print("tot.kcal: ");
  lcd.print (totcal);
  pesostabile = 1; // setta il semaforo di peso stabile, per consentire
  // la scelta dell'alimento
}
//
// ***** routine di pulizia della selezione corrente *****
// ***** clear current selection *****
//
void pulisci (void)
{
  lcd.clear ();
  pesoprec = 0; // azzera il peso memorizzato
  pesostabile = 0; // predispone il sistema ad una nuova pesatura
  semaforo = 0; // predispone il semaforo all'acquisizione del primo codice
}
```

## Arduino: il pesa calorie – the caloriescale

```
}
//
//***** routine di somma dell'ultimo ingrediente *****
//***** add last ingredient *****
//
void aggiungiallaricetta (void)
{
  ultalim = alim;           // memorizza il codice dell'alimento
  ultcat  = catalim;       // memorizza il codice della categoria
  ultpeso = peso;         // memorizza il peso dell'ingrediente che si sta aggiungendo
  ultcal  = cal;          // memorizza le calorie dell'ingrediente che si sta aggiungendo
  totcal  = totcal + cal;  // aggiunge le calorie correnti al totale della ricetta
  lcd.setCursor(0, 1);    // posiziona cursore a inizio seconda riga (carattere 0 e riga 1)
  lcd.print("ingrediente aggiunto");
  lcd.setCursor(0, 2);    // posiziona cursore a inizio terza riga (carattere 0 e riga 2)
  lcd.print("svuota il piatto "); // inserire spazi a destra sino a 20 caratteri
  lcd.setCursor(0, 3);    // posiziona cursore a inizio quarta riga (carattere 0 e riga 3)
  lcd.print(" "); // inserire spazi sino a 20 caratteri
  lcd.setCursor(0, 3);    // posiziona cursore a inizio quarta riga (carattere 0 e riga 3)
  lcd.print("tot.kcal: ");
  lcd.print (totcal);
  delay (3000);          // attende 3 secondi prima di continuare nella elaborazione
  pulisci ();           // predispose il sistema ad accettare un nuovo ingrediente
}
//
//***** routine di richiesta conferma eliminazione *****
// ***** confirm elimination *****
//
void chiediconferma(void)
{
  if (semaforo == 6)     // se e' stato chiesto il reset
  {
    lcd.clear ();        // pulisce lo schermo
    lcd.setCursor(0, 0); // posiziona cursore a inizio prima riga (carattere 0 e riga 0)
    lcd.print("vuoi cancellare "); // inserire spazi a destra sino a 20 caratteri
    lcd.setCursor(0, 1); // posiziona cursore a inizio seconda riga (carattere 0 e riga 1)
    lcd.print("tutto e ripartire? "); // erase all and restart?
    lcd.setCursor(0, 3); // posiziona cursore a inizio della terza riga (carattere 0 e riga 3)
    lcd.print("confermi? S/N"); // chiede conferma (confirm y/n (*/#))
  }
  else
  {
    semaforo = 2;        // predispose la ricezione del codice di conferma o di eliminazione
    lcd.clear ();        // pulisce lo schermo
    lcd.setCursor(0, 0); // posiziona cursore a inizio prima riga (carattere 0 e riga 0)
    lcd.print ("canc. ");
    lcd.print(ultpeso);  // evidenzia il peso dell'ultimo ingrediente aggiunto
    lcd.print ("g ");
    lcd.print (ultcal);  // evidenzia le calorie dell'ultimo ingrediente aggiunto
    lcd.print ("kcal");
    lcd.setCursor(0, 1); // posiziona cursore a inizio seconda riga (carattere 0 e riga 1)
    lcd.print(categoria [ultcat]); // evidenzia la categoria dell'ultimo ingrediente
    lcd.setCursor(0, 2); // posiziona cursore a inizio terza riga (carattere 0 e riga 2)
    lcd.print(alimento [ultalim]); // evidenzia il nome dell'ultimo ingrediente
    lcd.setCursor(0, 3); // posiziona cursore a inizio terza riga (carattere 0 e riga 3)
    lcd.print("confermi? S/N"); // chiede conferma
  }
}
//
//***** routine di selezione della categoria alimentare*****
// ***** select food category *****
//
void categorialimentare (void)
{
  semaforo = 1;          // predispose alla ricezione del secondo codice (alimento)
  catalim = tastocorrente; // memorizza la categoria alimentare
  lcd.setCursor(0, 1);   // posiziona cursore a inizio seconda riga (carattere 0 e riga 1)
  lcd.print(categoria [catalim]); // visualizza sul display (riga 1) la categoria alimentare
  lcd.setCursor(0, 2);   // posiziona cursore a inizio terza riga (carattere 0 e riga 2)
  lcd.print("codice alimento?"); // chiede l'inserimento del codice di alimento - food code?
}
//
//***** routine di selezione dell'ingrediente *****
// *****select food *****
//
void ingrediente (void)
{
  semaforo = 3;
```

## Arduino: il pesa calorie – the caloriescale

```
lcd.setCursor(0, 2); // posiziona cursore a inizio terza riga (carattere 0 e riga 2)
alim = (catalim * 10) + tastocorrente; // Calcola la posizione dell'alimento in tabella
lcd.print(alimento [alim]); // espone la descrizione dell'alimento
lcd.setCursor (0, 0); // posiziona il cursore sulla prima riga
lcd.print (" "); // inserire spazi sino a 20 caratteri
lcd.setCursor (0, 0); // posiziona il cursore sulla prima riga
lcd.print (peso);
lcd.print ("g - kcal: ");
calalim = calorie[alim];
comodocalcolo = peso * calalim; // calcolo intermedio - intermediate result
Serial.print (peso);
Serial.print (" - ");
Serial.print (calalim);
Serial.print (" - ");
Serial.print (comodocalcolo);
Serial.print (" - ");
cal = comodocalcolo / 100; // calcola le calorie dell'ingrediente sul piatto
lcd.print (cal);
lcd.setCursor (0, 3); // posiziona il cursore sulla quarta riga
lcd.print ("a:canc. / b:aggiungi"); // "a" delete "b" adds
}
//
//***** routine di conferma dell'eliminazione dell'ultimo ingrediente*****
// ***** confirm elimination last ingredient *****
//
void confermaeliminazione (void)
{
  if (semaforo == 2)
  {
    totcal = totcal - ultcal; //sottrae l'ultimo ingrediente al totale calorie
    ultpeso = 0; // azzera ultimo peso
    ultcal = 0; // azzera ultime calorie
    lcd.clear (); // pulisce lo schermo
    lcd.setCursor(0, 0); // posiziona cursore a inizio prima riga (carattere 0 e riga 0)
    lcd.print ("ingrediente "); // ingredient
    lcd.setCursor(0, 1); // posiziona cursore a inizio seconda riga (carattere 0 e riga 1)
    lcd.print("cancellato"); // erased
    lcd.setCursor(0, 3); // posiziona cursore a inizio quarta riga (carattere 0 e riga 3)
    lcd.print("nuovo tot.kcal: ");
    lcd.print (totcal);
    delay (3000); // attende tre secondi
  }
  else
  {
    //
    // *****pone il sistema in attesa di una nuova ricetta*****
    // *****wait for a new recipe *****
    //
    totcal = 0; // azzera il totale della ricetta
    lcd.clear ();
    lcd.print ("bilancia pronta ");
    lcd.setCursor(0, 1); // posiziona cursore a inizio seconda riga (carattere 0 e riga 1)
    lcd.print("per una nuova");
    lcd.setCursor(0, 2); // posiziona cursore a inizio terza riga (carattere 0 e riga 2)
    lcd.print("ricetta");
    lcd.setCursor(0, 3); // posiziona cursore a inizio quarta riga (carattere 0 e riga 3)
    lcd.print("tot.kcal: ");
    lcd.print (totcal);
    delay (3000); // attende 3 secondi
  }
  pulisci ();
}
//
//
void setup()
{
  Serial.begin (9600); // attiva monitor seriale per comunicazioni di debug
  lcd.begin(20, 4); // inizializza il display (20 caratteri per 4 righe)
  lcd.backlight(); // illumina lo sfondo del display
  lcd.clear(); // pulisce il display
  lcd.setCursor(0, 0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print("calcolo della tara"); // tare
  lcd.setCursor(0, 1); // posiziona il cursore all'inizio della seconda riga (carattere 0 e riga 1)
  lcd.print("non porre nulla"); // don't put anything
  lcd.setCursor(0, 2); // posiziona il cursore all'inizio della terza riga (carattere 0 e riga 2)
  lcd.print("sulla bilancia...."); // on plate...
  lcd.setCursor(0, 3); // posiziona il cursore all'inizio della quarta riga (carattere 0 e riga 3)
```

## Arduino: il pesa calorie – the caloriescale

```
bilancia.set_scale(2045); // valore di scala, definito tramite il programma inserito nell'esercizio
41 - Scale value, see exercise 41 to compute it
bilancia.tare(20); // il peso attuale e' considerato tara
lcd.clear(); // pulisce il display
lcd.print("bilancia pronta"); // system ready
}
//
//
void loop()
{
  if (pesostabile == 0) // se si deve procedere ad una pesatura
  {
    iniziopesostabile = millis (); // memorizza il momento di inizio della pesatura
    pesatura (); // lancia la routine di pesatura
  }
  else
  {
    tastobattuto = keypad.getKey(); // acquisisce il valore ascii del tasto battuto
    // (gia' trascodificato tramite la tabella chiave)
    if (tastobattuto) // se e' stato premuto un tasto
    {
      if (tastobattuto == 'a') // richiesta di eliminazione della selezione corrente
        pulisci ();
      if ((semaforo == 3) && (tastobattuto == 'b')) // richiesta di aggiunta alla ricetta
        aggiungiallaricetta ();
      if ((totcal > 0) && (tastobattuto == 'c')) // richiesta di cancellazione dell'ultimo
        ingrediente inserito nella ricetta
        chiediconferma ();
      if (tastobattuto == 'd') // richiesta di reset della ricetta
      {
        semaforo = 6; // imposta il semaforo di reset della ricetta
        chiediconferma();
      }
      if (((semaforo == 2) || (semaforo == 6)) && (tastobattuto == '*')) // se viene confermata
        l'eliminazione - if erase is confirmed
        confermaeliminazione ();
      if (((semaforo == 2) || (semaforo == 6)) && (tastobattuto == '#')) // se viene annullata la
        eliminazione - if erase is deleted
        pulisci ();
      if ((tastobattuto >= '0') && (tastobattuto <= '9'))
      {
        tastocorrente = tastobattuto; // memorizza in una zona di lavoro numerica il valore ascii
        del tasto battuto
        tastocorrente = tastocorrente - 48; // trasforma il valore ascii in un valore numerico da 0
        a 9
        if (semaforo == 0) // se si attendeva una categoria alimentare
          categorialimentare ();
        else if (semaforo == 1) // se si attendeva un codice di alimento
          ingrediente (); // lancia la routine di acquisizione del codice di alimento - ask food
        code
      }
    }
  }
}
```