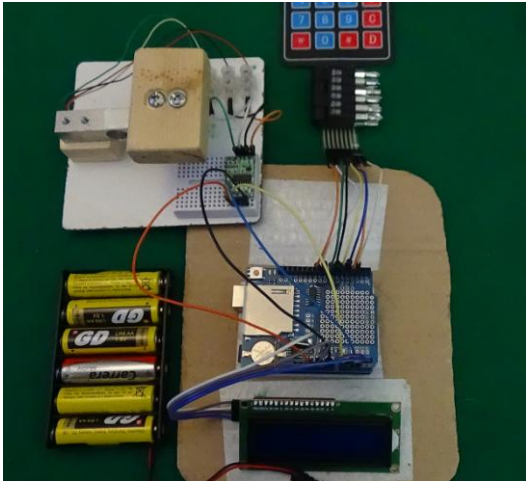


Z1- registratore di peso su scheda SD – weight recorder on SD card (some notes at end of this section)



Questo impianto pesa un oggetto, ne acquisisce il codice da tastiera e registra, su secure digital, data, ora, peso e codice.

Per quanto riguarda gli aspetti costruttivi, alcune indicazioni riguardanti la costruzione della bilancia e l'uso della tastiera e del display a cristalli liquidi, sono presenti nelle schede:

- [41 bilancia digitale](#)
- [42 tastiera e display](#)

In particolare, Nella scheda 41 sono presenti sia il programma che le indicazioni per il calcolo del **valore di scala**, specifico di ogni sensore di peso. Questo valore, una volta calcolato, deve essere poi inserito nel

programma prima della sua compilazione (vedi piu' sotto le note in rosso, nella sezione dedicata al programma). Una volta calcolato il valore di scala e' normalmente necessario procedere ad un lavoro di calibratura fine, sperimentando diversi valori (di poco maggiori o inferiori al valore calcolato) fino a quando la bilancia non arriva ad esporre, con precisione, il peso di un campione di riferimento.

L'impianto e' stato costruito con componenti reperiti sui mercati cinesi. In fase di test sono state rilevate alcune difficolta' di funzionamento della stazione di lettura/scrittura SD e del sensore di peso. Per quanto riguarda la stazione di lettura/scrittura SD le difficolta' sono state superate con l'utilizzo di una scheda arduino originale mentre per la bilancia si e' reso necessario ricalcolare il valore di scala.

Sotto l'aspetto operativo Arduino, al momento dell'accensione deriva da RTC la data e l'ora correnti, calcola la tara della bilancia, verifica se la stazione di lettura/scrittura e' operativa e quindi si pone in attesa di un codice oggetto (se la stazione di lettura/scrittura non funziona, segnala il problema ma poi si mette comunque in attesa di un codice oggetto).

L'operatore batte sulla tastiera il codice oggetto (max tre cifre) che, se inferiore a tre cifre, deve essere seguito dalla pressione del tasto "#" (il tasto "#" assume la funzione del tasto "enter").

Una volta ricevuto il codice oggetto lo controlla su di una tabella interna. Se il codice non e' in tabella espone, per tre secondi, una dicitura di avviso ma poi lo accetta comunque, a meno che non venga premuto il tasto "a" annullamento.

Attende quindi che venga posto un oggetto sul piatto della bilancia e, appena viene posato procede alla sua pesatura. Quando il peso e' stabilizzato e se l'oggetto pesa piu' di 5 grammi, registra su secure digital alcuni dati (data ora, codice oggetto e peso in grammi) e ricomincia il ciclo, ponendosi in attesa di un nuovo codice oggetto.

I dati registarti sulla scheda secure digital possono poi essere letti su di un PC ed eventualmente importati su di un foglio excell per eventuali ulteriori elaborazioni.

Il file sulla SD, in formato .txt puo' essere letto, modificato o cancellato utilizzando wordpad o le normali funzioni di gestione file di windows. Per limitare i problemi di registrazione dei dati su SD e' necessario, ogni qualvolta si utilizza la SD su un PC, chiuderla mediante la normale procedura prevista da windows per la chiusura e l'estrazione delle schede di memoria (procedura di rimozione sicura dell'hardware).

	A	B	C	D	E	F
1	15/01/2017	11.59.51	56	535		
2	15/01/2017	12.00.27	237	529		
3	15/01/2017	12.08.02	7	548		
4	15/01/2017	12.08.23	64	71		
5	15/01/2017	12.08.53	999	46		
6	15/01/2017	12.10.00	7	525		
7	15/01/2017	12.10.20	64	68		
8	15/01/2017	12.10.44	999	47		
9	15/01/2017	12.11.49	7	547		
10	15/01/2017	12.54.40	7	547		
11	15/01/2017	12.56.13	85	4		
12	15/01/2017	13.00.20	7	540		
13	15/01/2017	13.00.37	55	6		
14	15/01/2017	13.07.53	7	549		
15	15/01/2017	13.08.24	64	47		
16	15/01/2017	13.08.39	999	70		
17	15/01/2017	13.09.19	7	549		
18	15/01/2017	13.10.01	64	48		
19	15/01/2017	13.10.21	999	73		
20	15/01/2017	13.10.38	55	74		
21	15/01/2017	13.11.00	7	527		
22	15/01/2017	13.11.16	64	73		
23	15/01/2017	13.11.34	55	41		
24						

Arduino: registratore di peso su SD– weigh recorder on Secure Digital

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

- ds3232RTC.h reperibile in <https://github.com/JChristensen/DS3232RTC>
- Time.h reperibile in <https://github.com/PaulStoffregen/Time>
- LiquidCrystal_I2C.h reperibile in <https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads>
- Hx711 reperibile in <https://github.com/bogde/HX711>

Per installare le librerie e' necessario seguire la procedura illustrata nei precedenti progetti, e sintetizzabile in:

- download della libreria in formato complesso
- 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)

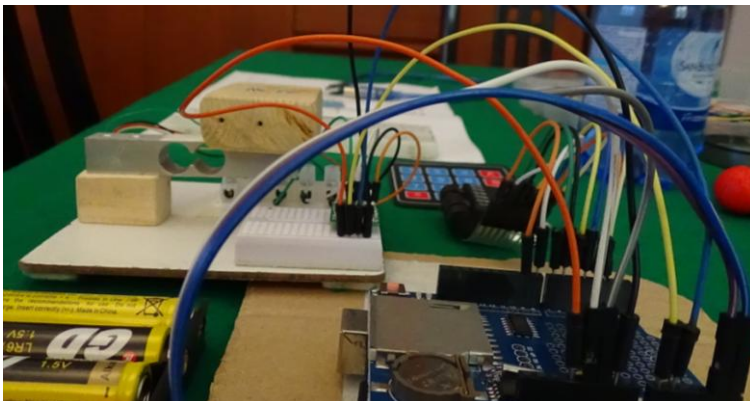
Sempre prima di procedere alla compilazione bisogna anche installare la libreria "keypad", presente ma non installata, nel library manager di Arduino (ide->sketch->include library->manage library->keypad).

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



This system weighs an object, acquires its item code from keyboard and writes on secure digital: date, time, weight and code.

As for the construction aspects, some indications (in english too) concerning the digital scale construction and use of keyboard and LCD display, can be found in:

- [41 digital scale](#)
- [42 keyboard and display](#)

In project 41, you can find both the program and the signs for the **scale value** calculation, specific to each sensor. This value, once determined, must be inserted in program prior to its compilation (see under the red notes, in the program section). Once calculated the **scale value**, is normally necessary a fine calibration work, experimenting different values (slightly more or less than the calculated value) up to when the scale does not exhibit the weight of the reference sample

The plant was built with components sourced on chinese markets. In testing were detected some difficulties in data logger and load cell use. As for the data logger the difficulties have been

Arduino: registratore di peso su SD- weigh recorder on Secure Digital

overcome using an original Arduino board while for the load cell was necessary to recalculate the scale value.

Under operational aspect, at starting time arduino derives current date and time from RTC, calculates the balance tare, checks if datalogger can work and then waits for an item code (if the datalogger does not work, reports the problem but then puts itself waiting for an item code). The operator inputs the item code on keypad; maximum three digits; if less, they must be followed by a "#".

Upon receipt the item code, checks it on an internal array. If code is not present in array, shows a warning indication for three seconds, and then accepts it anyway, unless you presses button "a", to cancel it.

Hence waits for an object placed on scale plate. Weighs the object on plate and when stabilized (and if is more than 5 grams), records some data on Secure Digital: (date, time, item code and weight in grams) and the cycle begins again, putting itself on hold for a new item code.

The recorded data can then be read on a PC and possibly imported on an excel sheet for any possible further processing.

The file on sd, in .txt format, can also be read, modified or deleted using wordpad or normal windows file management functions. To limit problems is necessary, every time you use the SD on a PC, close it using the standard procedure for hardware safe removal.

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

- ds3232rtc.h found [here](#).
- time.h found [here](#)
- LiquidCrystal_I2C.h found [here](#)
- Hx711 found [here](#)

Before compilation must also installed library "keypad", present but not installed, in the Arduino library manager (ide-> sketch-> includes Library-> manage Library-> keypad).

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

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

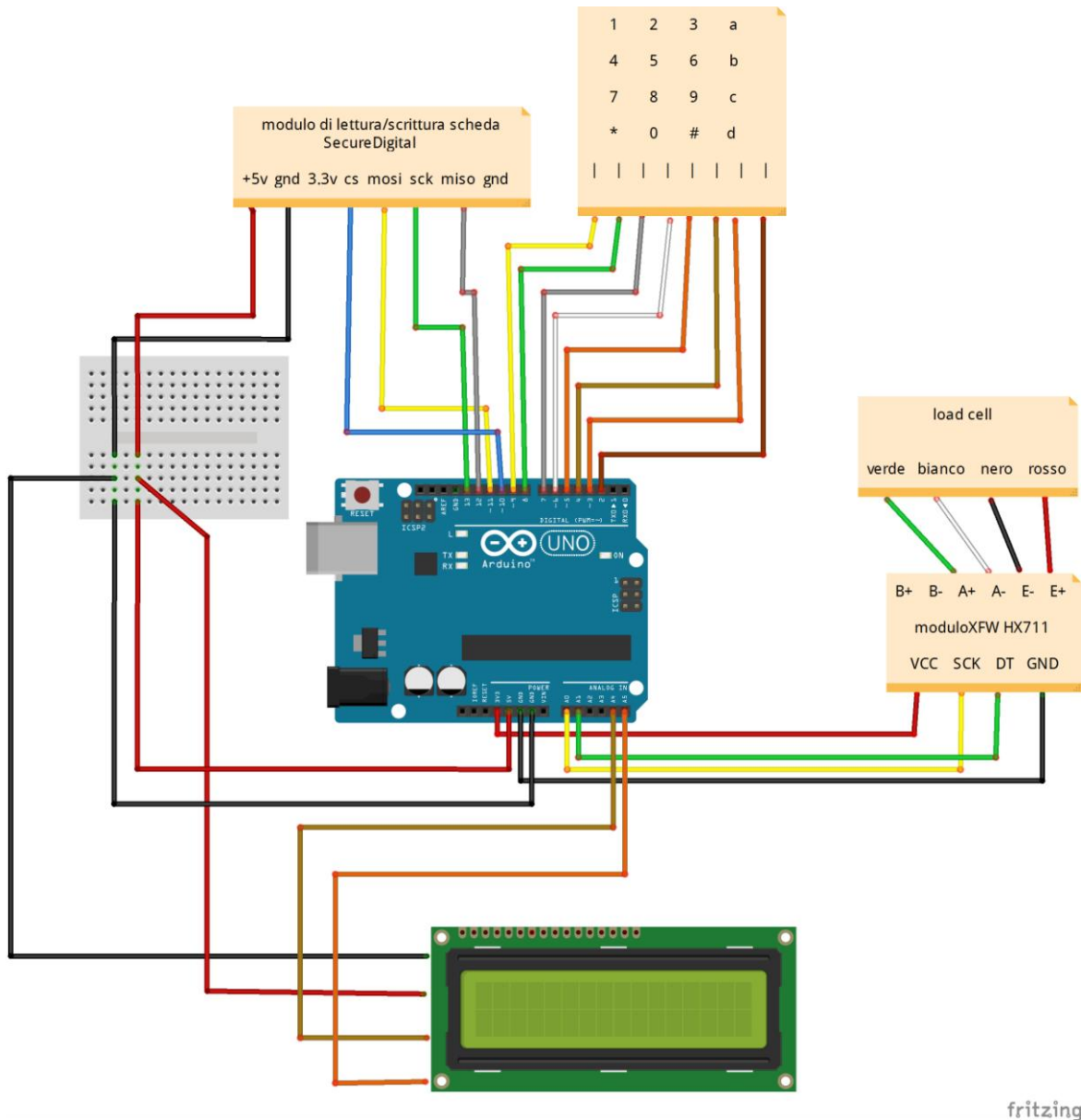
- Un modulo datalogger composto da una stazione di lettura scrittura di schede sd e da un modulo RTC ds3231 oppure da una stazione di lettura scrittura e da un sistema RTC separati (in quest'ultimo caso il modulo rtc deve essere collegato al bus scl/sda (i due pin a sinistra del pin AREF) - A data logger module, composed by a read-write station for secure

Arduino: registratore di peso su SD– weigh recorder on Secure Digital

digital card and by a ds3231 RTC module or a read-write station and a separate RTC system (in the latter case the RTC module must be connected to scl/sda bus (two pins on the left of AREF pin))

- Una keypad da 16 tasti
- Load cell e relativo driver HX711
- Display lcd tipo 1602 dotato di driver I2C

Schema



fritzing

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.
 *
 * Utilizzo di un sensore di peso, di un tastiera, di un display 1602, e di un datalogger per un
 * impianto di pesatura e registrazione
 *
 * Prima di compilare il programma scaricare ed installare, se non gia' fatto, le seguenti librerie:
 * - gestione del driver XH711: https://github.com/bogde/HX711
 * - driver I2C del display: https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
 */
```

Arduino: registratore di peso su SD– weigh recorder on Secure Digital

```
* - libreria keypad.h, presente nel "library manager" dell'IDE (da installare tramite il library
* manager)
*
* connessioni del modulo H711:
* Hx711.DO - porta #A1
* Hx711.SCK - porta #A0
*
* connessioni del sensore di peso:
* rosso = E+
* nero = E-
* bianco = A-
* verde = A+
*
* connessioni della tastiera:
* porte da 2 a 9, in linea con le porte del connettore:
* tenendo i pulsanti rivolti verso l'alto, collegare il primo connettore di destra
* alla porta 2, il secondo alla 3 e così via, fino all'ottavo (primo da sinistra)
* da collegare alla porta 9
*
* collegamento del lettore/registratori sd con RTC (real time clock)
*
* .....arduino uno.....Arduino mega
* GND .....non usato.....non usato
* MISO .....12 .....51
* SCK .....13 .....52
* MOSI .....11 .....50
* CS..... 10 .....53
* 3.3v .....non usato .....non usato
* GND .....GND .....GND
* +5V .....5v .....5v
*
* collegamento del display lcd con driver I2C
*
* GND.....GND .....GND
* VCC .....5v .....5v
* SDA .....A4 .....20
* SCL .....A5 .....21
*
*-----
* Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
*
* Use a load cell, a keyboard, a display 1602, and a data logger
* for a weighing and recording system
*
* Before compiling program must be entered the "scale value" determined following
* the notes present in project 41:
* http://giocarduino.altervista.org/e41-sensore-di-peso-bilancia-digitale.pdf
*
* connections of H711 module to Arduino:
* DO = analogic pin A1
* SCK = analogic pin A0
*
* Connections load cell to HX711:
* red = E+
* black = E-
* white = A-
* green = A+
*
* Note: if system proposes negative values, reverse the connection of white and green wires
*
* Connection on SD appliance:
* .....arduino uno.....Arduino mega
*
* GND .....not used.....not used
* MISO .....12 .....51
* SCK .....13 .....52
* MOSI .....11 .....50
* CS..... 10 .....53
* 3.3v .....not used .....not used
* GND .....GND .....GND
* +5V .....5v .....5v
*
* Connection on display 1602 with I2C driver
*
* GND.....GND .....GND
* VCC .....5v .....5v
* SDA .....A4 .....20
* SCL .....A5 .....21
```

Arduino: registratore di peso su SD– weigh recorder on Secure Digital

```
*
*
* keypad connections:
* rows pins (first 4 connector on the left): pins 9,8,7 and 6
* columns pins (last 4 connectors on the righth): pins 5,4,3,and 2
* -----
*/
#include "HX711.h"           // https://github.com/bogde/HX711
#include <Keypad.h>         // presente nel library manager dell'IDE
#include <Wire.h>           // libreria wire, presente di default nell'IDE
#include <DS3232RTC.h>      // http://github.com/JChristensen/DS3232RTC
#include <TimeLib.h>       // http://playground.arduino.cc/Code/Time
#include <LiquidCrystal_I2C.h> // https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
#include <SPI.h>           // libreria spi, presente di default nell'IDE
#include <SD.h>            // libreria SD, presente di default nell'IDE
#define DOUT A1
#define CLK A0
File fileasd;
HX711 bilancia(DOUT, CLK);
// 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);
time_t t;
tmElements_t tm;
int digits = 0; // zona di memorizzazione dei minuti e dei secondi da esporre
// sul display lcd
int mese = 0; // zona di memorizzazione del mese corrente
int lavoro = 0; // zona di lavoro, utilizzata per calcoli intermedi
char tabdescr [7] [16] // tabella descrizioni da esporre su lcd
{
  {"enter item id:"}, // 0
  {"open sd ok"}, // 1
  {"sd ko"}, // 2
  {"item recorded"}, // 3
  {"cancel"}, // 4
  {"item on plate"}, // 5
  {"invalid code"} // 6
};
int codescr = 0; // codice del messaggio da esporre
char nomefile [10] = {"text.txt"}; // nome del file su sd
int tabcodici [31] = // tabella dei codici validi
{
  2, // tabella dei codici oggetto. devono essere sempre 30, non uno di piu', non uno di meno
  5, // item code array: must be always 30 numbers (form 0 to 999) no one more, no one less
  7,
  12,
  56,
  64,
  92,
  98,
  103,
  134,
  206,
  237,
  290,
  292,
  301,
  350,
  374,
  421,
  503,
  509,
  641,
  730,
  792,
  803,
  858,
  0,
  0,
  0,
  0,
  0,
  0,
};
byte trovato = 0; // semaforo per codice oggetto valido:
// 0 = codice oggetto non presente in tabella
// 1 = codice oggetto presente in tabella
char tastobattuto;
const byte righe = 4; // numero di righe della tastiera (quattro)
```

Arduino: registratore di peso su SD– weigh recorder on Secure Digital

```
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
  {'1', '2', '3', 'a'},
  {'4', '5', '6', 'b'},
  {'7', '8', '9', 'c'},
  {'*', '0', '#', 'd'}
};
byte pinrighe[righe] = {9, 8, 7, 6}; //porte cui devono essere connessi i pin delle
// righe (i primi quattro pin di sinistra) della tastiera
byte pincolonne[colonne] = {5, 4, 3, 2}; //porte cui devono essere connessi i pin delle
// colonne (gli ultimi quattro pin di destra) dellatastiera
//prossima riga: assegnazione dei parametri alla libreria keypad.h
Keypad keypad = Keypad( makeKeymap(chiave), pinrighe, pincolonne, righe, colonne );
long valore = 0; // valore immesso da tastiera
char tabnum [4]; // tabella di memorizzazione dei caratteri immessi
byte indtab = 0; // indice di scorrimento della tabella tabnum
byte semaforo = 0; //semaforo per la permanenza nella routine di ricezione valori:
// 0 = resta nella routine;
// 1 = esci dalla routine e prosegui nel lavoro;
// 2 esci dalla routine e ricomincia il loop
byte indice = 0; // indice utilizzato nel calcolo dei valori immessi
long moltiplicatore = 1; // moltiplicatore utilizzato nel calcolo del valore
int peso = 0; // zona di memorizzazione del peso corrente
int pesoprec = 0; // peso rilevato nella precedente pesata
int i = 0; // indice utilizzato in svariati punti del programma
//
// ***** routine di esposizione dei messaggi su dispaly lcd *****
// ***** message on lcd (first row) *****
//
void messaggio (void)
{
  lcd.clear ();
  lcd.print (tabdescr [codescr]);
}
//
//***** routine di esposizione di data, ora sul display lcd ****
// ***** time on lcd *****
//
void esponidata(void)
{
  // esposizione dei dati provenienti dal timer di arduino
  lcd.clear();
  lcd.setCursor(3, 0);
  lcd.print(day());
  lcd.print('/');
  lcd.print(month());
  lcd.print('/');
  lcd.print(year());
  lcd.setCursor(3, 1);
  lcd.print(hour());
  lcd.print(':');
  digits = (minute());
  printDigits(); // routine esposizione zeri non significativi
  lcd.print(':');
  digits = (second());
  printDigits(); // routine esposizione zeri non significativi
  delay (3000);
}
//
//****routine visualizzazione minuti e secondi comprensivi di zeri non significativi****
//
void printDigits()
{
  if (digits < 10)
    lcd.print('0');
  lcd.print(digits);
}
//
//***** routine di scrittura su secure digital *****
// ***** record on secure digital *****
//
void scrivi (void)
{
  filesd = SD.open("testo.txt", FILE_WRITE); //File in scrittura
```

Arduino: registratore di peso su SD– weigh recorder on Secure Digital

```
if (filesd)          // Se il file è stato aperto correttamente
{
  filesd.print(year());
  filesd.print("/");
  filesd.print(month());
  filesd.print("/");
  filesd.print(day());
  filesd.print(",");
  filesd.print(hour());
  filesd.print(":");
  filesd.print(minute());
  filesd.print(":");
  filesd.print(second());
  filesd.print(",");
  filesd.print(valore);
  filesd.print(",");
  filesd.print (peso);
  filesd.println ("");
  filesd.close(); // Chiude il file su sd
  codescr = 3;    // segnale scrittura completata
}
else
  codescr = 2;    // segnala errore apertura file
messaggio ();
esponisituazione();
delay (4000);
}
//
// ***** routine di acquisizione e verifica dati da tastiera *****
// ***** acquire and verify data form keypad *****
//
void veditastiera (void)
{
  lcd.setCursor (0, 1);
  valore = 0;
  semaforo = 0;
  for (indtab = 0; indtab <= 3; indtab++)
    tabnum[indtab] = 0;
  indtab = 0;
  while (semaforo == 0)    // loop di ricezione valori
  {
    tastobattuto = keypad.getKey();    // acquisisce il valore del tasto battuto (gia'
    trascodificato)
    if (tastobattuto)    // se e' stato battuto un tasto
    {
      lcd.print (tastobattuto);    // visualizza il tasto battuto sul display lcd
      if ((tastobattuto >= '0') && (tastobattuto <= '9'))
      {
        tabnum [indtab] = tastobattuto - 48; // memorizza il valore battuto, trasformato in un
// numero da 0 a 9
        indtab++;
      }
    }
    if ((tastobattuto == '#') || (indtab == 3))
    {
      // ***** trasforma in un unico numero i singoli numeri digitati *****
      // ***** transform in value each single digit inputted by keypad *****
      semaforo = 1;
      moltiplicatore = 1;
      for (indice = indtab; indice > 1; indice--)
        moltiplicatore = moltiplicatore * 10;
      for (indice = 0; indice <= indtab; indice++)
      {
        valore = valore + tabnum[indice] * moltiplicatore;
        moltiplicatore = moltiplicatore / 10;
      }
      //***** verifica validita del codice immesso *****
      // ***** item code validation *****
      trovato = 0;
      for (indice = 0; indice <= 30; indice++)
      {
        if (valore == tabcodici [indice])
          trovato = 1;    // codice trovato in tabella - item code founded in "tabcodici"
      }
      if (trovato == 0)    // valore non trovato in tabella
      {
        codescr = 6;    // "codice non valido" Invalid item code
        messaggio ();
        delay (4000);
      }
    }
  }
}
```


Arduino: registratore di peso su SD- weigh recorder on Secure Digital

```
    }
    codescr = 5; // poni un oggetto sul piatto
    messaggio ();
    esponisituazione ();
  }
  if (tastobattuto == 'a') // annulla comando e riparti dall'inizio - delete code and restart
    annulla ();
}
}
//
// ***** routine di annullamento dati battuti *****
// ***** delete code *****
//
void annulla (void)
{
  valore = 0;
  semaforo = 2; // riparte da inizio loop
  for (indtab = 0; indtab <= 3; indtab++)
    tabnum[indtab] = 0;
  codescr = 4; // segnala codice cancellato
  messaggio ();
  esponisituazione ();
  delay (3000);
}
//
//**** routine di esposizione della situazione corrente
// ***** show item code and current weight *****
//
void esponisituazione (void)
{
  messaggio ();
  lcd.setCursor (0, 1);
  lcd.print ("item:");
  lcd.print (valore);
  lcd.setCursor (9, 1);
  lcd.print ("gr:");
  lcd.print (peso);
}
//
//
void setup()
{
  Serial.begin (9600); // inizializza la comunicazione seriale
  lcd.begin(16, 2); // inizializza il display (16 caratteri per 2 righe)
  lcd.backlight(); // illumina lo sfondo del display
  lcd.setCursor(0, 0); // posiziona il cursore all'inizio della prima riga (carattere 0 e riga 0)
  lcd.print ("good day");
  delay (1500);
  lcd.clear ();
  lcd.setCursor(0, 0); // posiziona il cursore all'inizio della prima riga
  // (carattere 0 e riga 0)
  setSyncProvider(RTC.get()); // sincronizza il timer di Arduino con i dati presenti
  // sul modulo RTC
  lcd.clear ();
  if (timeStatus() != timeSet) // verifica se la sincronizzazione e' andata a buon fine
    lcd.print("clock not");
  else
    lcd.print("clock");
  lcd.setCursor (0, 1);
  lcd.print ("synchronized");
  delay (1500);
  esponidata ();
  lcd.clear ();
  lcd.print("tare verify"); // " tare verify"
  lcd.setCursor (0, 1);
  lcd.print("please wait..."); //
  bilancia.set_scale(805); // valore di scala, definito tramite il programma
  // inserito nella scheda descrittiva del progetto - Insert the "scale value"
  // computed by program in project 41 (see inital note)
  bilancia.tare(40); // il peso attuale e' considerato tara
  delay (1500);
  pinMode (10, OUTPUT); // il pin CS e' collegato alla porta 10
  if (!SD.begin(10)) // verifica la presenza della secure digital
  {
    codescr = 2; // segnala apertura KO - sd ko
  }
  else

```

Arduino: registratore di peso su SD- weigh recorder on Secure Digital

```
{
  codescr = 1;      // segnala apertura OK - open sd OK
}
messaggio ();
delay (2000);
}
//
//
void loop()
{
  semaforo = 0;
  codescr = 0;
  peso = 0;
  messaggio (); //espone il messaggio 0: "enter item id"
  veditastiera(); // attende dati da tastiera
  if (semaforo == 1) // se e' stato battuto un codice
  {
    while (((semaforo == 1) && peso <= 5 ) || (! (peso == pesoprec))) // verifica se il peso si e'
    stabilizzato - verify if weight is stabilized
    {
      pesoprec = peso;
      delay (200); // attende 2/10 di secondo per consentire la stabilizzazione della bilancia
      tastobattuto = keypad.getKey(); // acquisisce il valore dell'eventuale tasto battuto (gia'
// trascodificato)
      if (tastobattuto == 'a')
      {
        annulla (); // annulla la pesatura corrente
      }
      peso = bilancia.get_units(2), 4; // rileva il peso sul piatto
      esponisituazione (); // espone la situazione corrente
    }
  }
  if (semaforo == 1) // se il semaforo e' sempre ad 1 (se non e' arrivato l'annullamento)
  scrivi (); // salva i dati su secure digital - save on secure digital
}
```