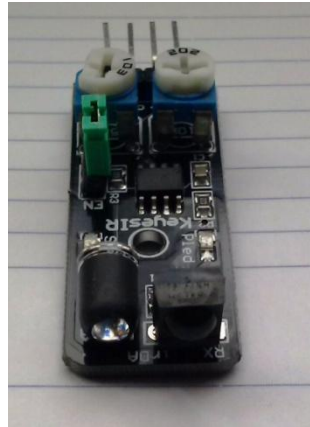


29- modulo avoidance: rilevatore di ostacoli a raggi infrarossi infrared obstacle detector (some notes at end of this section)



Il modulo keyes riportato in figura e' una basetta del costo di un paio di euro, sulla quale sono montati un generatore e un rilevatore di infrarossi, un paio di resistenze variabili, due led, un timer (usato probabilmente per modulare la sequenza dei segnali infrarossi), svariate resistenze e condensatori.

Le basi sulle quali opera il sensore sono abbastanza semplici. Il generatore lancia un fascio di segnali infrarossi che vengono riflessi da un eventuale ostacolo e poi intercettati dal modulo di ricezione.

I segnali riflessi perdono gran parte della loro forza per cui questo tipo di sensore puo' rilevare ostacoli posti ad una distanza massima di 20 centimetri (alcune specifiche parlano di 40 centimetri, ma prove pratiche hanno dimostrato di non poter andare oltre i 20 centimetri).

Agendo sulle resistenze variabili e' possibile diminuire la distanza massima di rilevamento fino a portarla ad un minimo di due centimetri.

La basetta fornisce, sul pin "out", un segnale di tipo digitale e non sembra in grado di fornire informazioni circa la possibile distanza dell'ostacolo.

L'utilizzo e' quindi ridotto alla semplice individuazione di ostacoli a distanza ravvicinata ed il modulo puo' probabilmente trovare applicazione pratica come contatore in impianti di media velocita'.

Lo sketch (il programma) allegato a questo esercizio vede il modulo utilizzato nella veste di un contatore di palline. Nel filmato conta alcune palline che corrono su dei binari passandogli davanti ad intervalli anche brevissimi e visualizza il conteggio su di un display lcd e sul monitor seriale.

Se non si dispone di un display lcd con driver LCMI 602, l'esercizio puo' comunque essere replicato limitando la visualizzazione del conteggio al solo monitor seriale.

Prima di procedere alla compilazione del programma deve essere installata, se non gia' presente, la seguente libreria:

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

Per installare la libreria 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)

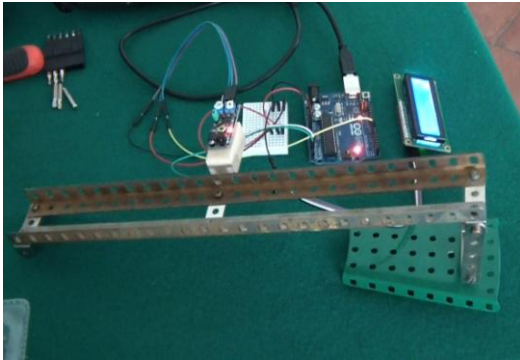
Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- [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 appliance shown in figure (a keysir module) is an infrared obstacle detector. On board are mounted a generator and an infrared detector, a pair of variable resistors, two LEDs, a timer (probably used to modulate the IR signal), various resistors and capacitors.

The basis on which the sensor operates are quite simple. The generator throws a beam of infrared signals that are reflected by any obstacle and then intercepted by the receiver module.

The reflected signals lose much of their strength so this type of sensor can detect obstacles at a maximum distance of 20 cm (about 40 cm from technical notes, but practical tests have shown that they can't go beyond 20 cm).

By acting on variable resistors you can decrease the detection range up to a minimum of two centimeters.

The device provides, on "out" leg, a digital signal and does not seem able to provide information about the obstacle distance. The use is therefore reduced to detection of close objects and can probably find practical application as a counter in medium speed systems.

The program sees the device used as a balls counter. In the movie some balls runs on tracks and passes in detector front, at even very short intervals. The counter is shown on an LCD and on serial monitor display.

If you do not have an LCD display, the project can still be replicated by displaying counter on serial monitor only.

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

- LiquidCrystal_I2C.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

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.

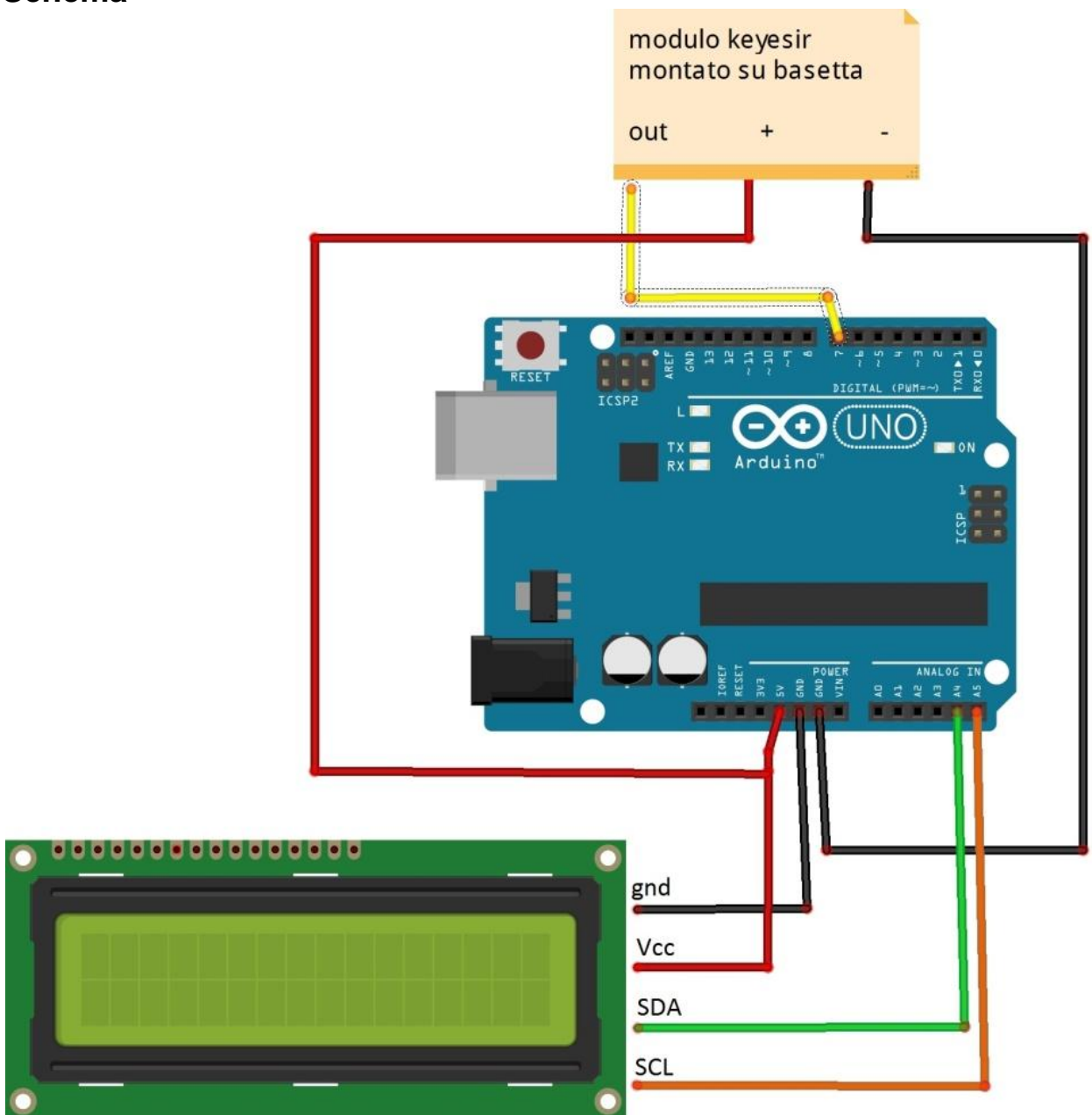
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

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 “avoidance” montato su basetta keys
- Un display lcd con driver I2C (facoltativo)
- Qualche pallina o comunque qualche oggetto da contare
- 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. Questo programma utilizza un modulo "avoidance" e cioe'
 * un modulo a raggi infrarossi, per contare oggetti che, passando davanti ad un generatore di
 * infrarossi, riflettono i raggi e li rendono intercettabili da un ricevitore. Lo schema vede un
 * modulo "avoidance", il cui pin "out" e' collegato alla porta 7 ed un display lcd con driver LCMI
 * 602 (con driver I2C) i cui pin SDA e SCL sono rispettivamente collegati alla porta A4 ed A5. Se
 * non si dispone di un display lcd il programma puo' comunque essere utilizzato cancellando le righe
 * contrassegnate dalla scritta "cancellare se no lcd"; poiche' il conteggio viene in ogni caso
 * esposto anche sul monitor seriale. (maggiori informazioni sull'utilizzo del display lcd con
 * driver LCMI 602 sono reperibili nell'esercizio 18 bis)
 *
 *-----
 * Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
 * This program uses a "avoidance" device to count objects passing in front of an infrared generator.
 * The objects, reflect rays and make them interceptable by a receiver. The schematic sees an
 * "avoidance" module, whose leg "out" is connected to the pin 7, and an LCD display with driver LCMI
 * 602 (a I2C driver) whose SDA and SCL legs are respectively connected to A4 and A5 pins. If you do
 * not have an LCD display, you can still be uses program by deleting lines marked by "cancellare se
 * no lcd" (in red on PDF format), since the count is in each case also exposed on the serial
 * monitor. (More information on using a LCD with I2C driver can be found in 18a project)
 *-----
 */

#include <Wire.h> // libreria wire presente, di default,
// nell'IDE
#include <LiquidCrystal_I2C.h> // cancellare se no lcd - 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); // cancella se no lcd
//
int contatore = 0; // variabile di memorizzazione del numero di oggetti contati
int semaforo = 0; // variabile che funge da semaforo, utilizzata per evitare di contare
// piu' volte il medesimo oggetto
int statokeyesir = 0; // variabile di memorizzazione dello stato (LOW oppure HIGH) del
// modulo keyesir
#define keyesir 7 // il pin "out" del modulo keyesir e' collegato alla porta 7
//
void setup()
{
  lcd.begin(16, 2); // cancellare se no lcd
  Serial.begin(9600); // inizializza il monitor seriale
  pinMode (keyesir, INPUT); // il sensore viene definito come unita' di input
}
//
void loop()
{
  statokeyesir = digitalRead (keyesir); // rileva lo stato del modulo
  if (statokeyesir == HIGH) // se non ci sono ostacoli
    semaforo = 0; // spigne il semaforo in modo da predisporre alla conta
  // del prossimo ostacolo
  if ((semaforo == 0) && (statokeyesir == LOW)) // se c'e' un nuovo ostacolo non conteggiato
  {
    contatore = contatore++; // conteggia l'ostacolo
    semaforo = 1; // considera conteggiato l'ostacolo

    // ***** display del conteggio su lcd *****
    lcd.setCursor(0, 0); // cancellare se no lcd
    lcd.print("num. palline: "); // cancellare se no lcd
    lcd.print(contatore); // cancellare se no lcd

    // ***** display del conteggio sul monitor seriale *****
    Serial.println(contatore);
  }
}

```