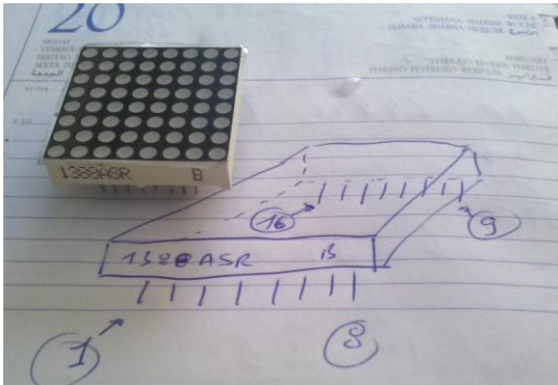


## 15 - matrice 8x8: scrivi e disegna - write and draw on an 8x8 led matrix (some notes at end of this section)



Questo esercizio utilizza il medesimo hardware (schema e componenti) del precedente esercizio 14, con cui condivide anche una parte di programma. In questa nuova versione sono presenti delle routine mediante le quali possono essere visualizzati caratteri e disegni realizzabili in una matrice da 8 X 8 punti. Il programma consente non solo di visualizzare i caratteri o i disegni in sequenza, ma anche di farli scorrere, come normalmente succede con le matrici led di maggiori dimensioni.

Sotto il profilo tecnico la gestione di ogni singolo led e' resa possibile dal fatto che ogni piedino pilota un'intera riga o un'intera colonna del display. Come gia' detto nel precedente esercizio, i piedini che pilotano le righe sono anodi (devono quindi essere collegati ad un polo positivo) mentre quelli che pilotano le colonne sono catodi (negativi). Il singolo led, all'incrocio di ogni riga e colonna, si accende quindi solo quando la polarita' e' positiva sulla riga e negativa sulla colonna mentre resta spento in ogni altra combinazione di casi.

Il programma gestisce ovviamente un led per volta ed il trucco adottato per gestire i 64 led in maniera apparentemente contemporanea si basa sulla persistenza della luminosita' dei led. La persistenza e' tale per cui Arduino puo' pilotare, in sequenza, ognuno dei led e giungere alla gestione del sessantaquattresimo led quando la luce del primo e' ancora visibile.

Unendo quindi la persistenza della luminosita', la velocita' di arduino e la corretta polarizzazione delle righe e delle colonne, e' possibile gestire ogni singolo led facendolo illuminare per un tempo minimo ma sufficiente a mantenerlo luminoso sino al successivo ciclo di refresh (fino alla successiva illuminazione).

Nel programma ogni carattere o disegno e' rappresentato in una tabella da 65 bytes disegnata come uno schema da 8 righe per 8 colonne (il primo byte non viene considerato). Ogni byte rappresenta un led e se nella tabella viene posto a 1, il relativo led si illuminera' quando la tabella verra' presa in carico dalla routine di visualizzazione.

Esempio di tabella per il disegno del carattere M maiuscolo e di tabella da inserire in programma

	<pre>int carM [65] = {   99, // il primo campo, con valore "99" e' un filler   1,0,0,0,1,0,0,0,   1,1,0,1,1,0,0,0,   1,1,0,1,1,0,0,0,   1,0,1,0,1,0,0,0,   1,0,1,0,1,0,0,0,   1,0,0,0,1,0,0,0,   1,0,0,0,1,0,0,0,   1,0,0,0,1,0,0,0, };</pre>
--	---

Il programma dopo aver inizializzato la matrice (accendendo e spegnendo, in sequenza, riga dopo riga, tutti i led) utilizza la routine di scorrimento del testo per far scorrere la parola "Rimini" scritta in corsivo, seguita dal disegno di un cuore che si ingrandisce e batte.

**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](mailto:giocarduino@libero.it)

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



In this project we uses the same hardware (schematics and components) of the previous project (# 14), with which it also shares a part of program.

In this new version is present a code by means of which can be displayed characters and drawings in a matrix of 8 x 8 points. The program not only allows displaying, but also scrolling, as normally happens with LED matrices. From a technical perspective the management of each individual LED is made possible because every pin drives an entire matrix row or column. As already mentioned in the previous project, the rows pins are anodes (must be connected to a positive pole) while columns pins are cathodes (negative). Each led turn on only when there is a correct polarity on row and column while it remains turned off in any other combination of cases.

Program turns on leds, one by one, and the trick adopted to handle 64 LEDs seemingly contemporary is based on persistence of LEDs brightness. The Arduino's speed and LED's persistence is such that Arduino manages each LED individually and arrive to manage the sixty-fourth LED, when the light of the first is still visible.

Then merging the persistence of brightness, the Arduino's speed and the correct polarization of rows and columns, is possible to manage each led, illuminating it for a minimum time, but enough to keep it bright until the next refresh (up to next lighting).

In program each character or design is represented by 65 bytes in a table drawn as a pattern of 8 rows by 8 columns (the first byte is not considered). Each byte is a LED and if in the table is set to 1, the corresponding LED on matrix will be lighted when table will be used from the display routine.

Down here is represented the "M" character. On the left as it appears on led matrix and on the right as it appear in program table

	<pre>int carM [65] = {   99, // il primo campo, con valore "99" e' un filler   1,0,0,0,1,0,0,0,   1,1,0,1,1,0,0,0,   1,1,0,1,1,0,0,0,   1,1,0,1,1,0,0,0,   1,0,1,0,1,0,0,0,   1,0,1,0,1,0,0,0,   1,0,0,0,1,0,0,0,   1,0,0,0,1,0,0,0,   1,0,0,0,1,0,0,0, };</pre>
--	--

The program after matrix initializing (turning on and off all leds, in sequence) scroll the word "Rimini" written in italics, followed by an heart design, that enlarges and beats.

**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.

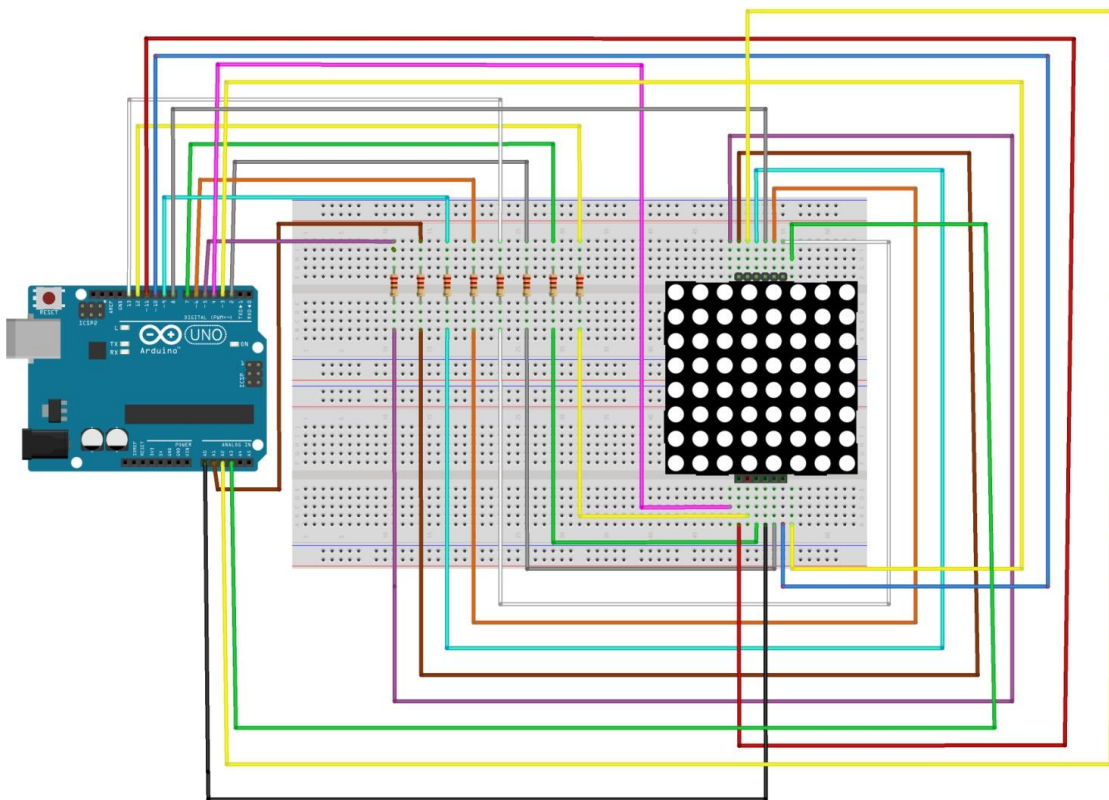
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

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

- 2 breadboard
- 8 resistenze da 220 ohm
- 1 matrice led 8x8
- 24 cavetti

## Schema



Per via dell'elevato numero di connessioni lo schema appare piuttosto complesso. E' probabilmente molto piu' semplice effettuare i collegamenti seguendo le indicazioni fornite dalla tabella di correlazione tra i piedini della matrice e le porte di Arduino ricordando ovviamente di frappare una resistenza tra i piedini che governano le colonne (catodi) e le relative porte.

Due to the high number of connections, schematic appears rather complex. Probably is much more easy to make connections following the instructions provided in correlation table between the matrix legs and Arduino's pins, remembering, of course, to interpose a resistor between pins that govern columns (cathodes) and matrix's legs.

## Arduino: matrice led 8x8, scrivi e disegna - write and draw on 8x8 led matrix

Tabella di correlazione tra i piedini della matrice led e le porte di Arduino

Piedino della matrice 8x8 Matrix leg (see figure on top)	Riga pilotata (anodi) Row managed	Colonna pilotata (catodi) Column managed	Porta di connessione di Arduino Arduino's pin
1	5		D4
2	7		D11
3		2	D12
4		3	D7
5	8		A0 (14)
6		5	D2
7	6		D10
8	3		D3
9	1		A3 (17)
10		4	D13
11		6	D6
12	4		D8
13		1	D9
14	2		A2 (16)
15		7	A1 (15)
16		8	D5

Il piedino numero 1 e' il primo piedino in basso a sinistra della matrice led e gli altri sono numerati in sequenza, procedendo in senso antiorario. Ne deriva che il sedicesimo piedino e' l'ultimo in alto a sinistra (vedi anche il disegno presente nel precedente esercizio 14).

Leg number 1 is the first leg at the bottom left of LED array and the others are numbered in sequence, proceeding counterclockwise. It follows that the sixteenth leg is the last top left (see also picture at top).

## 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. Il programma dopo aver inizializzato la matrice
 * (accendendo e spegnendo, in sequenza, riga dopo riga, tutti i led) utilizza la routine di
 * scorrimento del testo per far scorrere la parola "Rimini" scritta in corsivo, seguita dal disegno
 * do un cuore che ingrandisce e batte.
 *
 *-----
 * Warning: cut&paste from PDF to IDE loses formatting. to restore it press CTRL + T.
 * The program after matrix initializing (turning on and off all leds, in sequence) scroll the word
 * "Rimini" written in italics, followed by an heart design, that enlarges and beats.
 *-----
 */
int comodo [65]; // area di lavoro utilizzata dalla routine di esposizione della matrice
int carcomodo [65]; // area di lavoro utilizzata dalla routine di gestione dello scorrimento testo
int refr = 0; // indice utilizzato per il refresh del display
int z = 0; // indice utilizzato per spostare in tabella comodo i dati della matrice da visualizzare
int r = 0; // indice utilizzato per la scansione delle righe nella routine di gestione della matrice
int c = 0; // indice utilizzato per la scansione delle colonne nella routine di gestione matrice
int i = 0; // indice utilizzato in diverse parti del programma per pilotare righe e colonne
int colonna [8]; // tabella di memorizzazione della colonna da inviare alla routine di scorrimento
int semaforo = 0; // deviatore utilizzato per fornire il segnale di "colonna da visualizzare"
int r1 = 0; // indice di riga nella routine "esponi"
int c1 = 0; // indice di colonna nella routine "esponi"
int r2 = 0; // indice di riga nella routine "visualizzacolonna"
int c2 = 0; // indice di colonna nella routine "visualizza colonna"

/* Nella seguente tabella viene definita la correlazione tra i piedini della matrice (rappresentati
dall'indice utilizzato per scorrere la tabella) e la connessione alle porte di arduino: dalla porta
2 alla 17, considerando come porte 14,15, 16 e 17 le porte analogiche A0, A1,A2,e A3. Il primo
elemento della tabella (99) e' presente solo al fine di occupare un posto e consentire all'indice di
partire dalla colonna 1. Leggendo la tabella si evince che il piedino 1 e' collegato alla porta 5,
il piedino 2 alla porta 4, il piedino 3 alla porta 3...
 *
The following table defines the correlation between the matrix legs (represented by the index used
to scroll the table) and Arduino's pins: from 2 to 17, considering how ports 14,15, 16 and 17 analog
ports A0, A1, A2, and A3. The first element of the table (99) is present only in order to occupy a
```

## Arduino: matrice led 8x8, scrivi e disegna - write and draw on 8x8 led matrix

```
place and allow the index to start from column 1. Table shows that leg 1 is connected to the pin 5,
the leg 2 at pin 4, leg 3 to pin 3 ...
*/
int pins[17]= { 99, 5, 4, 3, 2, 14, 15, 16, 17, 13, 12, 11, 10, 9, 8, 7, 6};

/* nella seguente tabella viene definita la correlazione tra le colonne (dalla 1 alla 8) e le porte
di Arduino cui sono collegate. Guardando la precedente tabella si puo' comprendere che la colonna 1
e' collegata alla porta definita da pins[13] e cioe' alla porta 9; la colonna 2 e' collegata al
pins[10] e cioe' porta 12....
*
The below table define the correlation between matrix columns (from 1 to 8) and the pins of
which are connected. Looking at the above table on can understand that the column 1 is connected to
value in element #13 in pins table (that contain 9); the column 2 is connected to the value of
element 10 in pins table (value 12).... */
int cols[8] = {pins[13], pins[10], pins[15], pins[9], pins[4], pins[16], pins[6], pins[1]};

/* nella seguente tabella viene definita la correlazione tra le righe (dalla 1 alla 8) e le porte di
arduino cui sono collegate. Guardando la precedente tabella si puo' arguire che la riga 1 e'
collegata alla porta definita dal pins[8] e cioe' alla porta 17 (A4); la riga 2 e' collegata al
pins[7] e cioe' porta 16 (A3)....
*
The below table define the correlation between matrix rows (from 1 to 8) and the pins of
which are connected. Looking at the above table on can understand that the row 1 is connected to
value in element #8 in pins table (that contain 17); the row 2 is connected to the value of
element #7 in pins table (value 16).... */
int rows[8] = {pins[8], pins[7], pins[3], pins[14], pins[2], pins[12], pins[11], pins[5]};
//
// matrice di base per il disegno dei caratteri - matrix to draw characters
// int carBase [65] = {
// 99, // il valore "99" e' un filler, usato per riempire il primo (inutile) campo della tabella
// 0,0,0,0,0,0,0,0,
// 0,0,0,0,0,0,0,0,
// 0,0,0,0,0,0,0,0, // matrice di base per il disegno dei caratteri
// 0,0,0,0,0,0,0,0,
// 0,0,0,0,0,0,0,0,
// 0,0,0,0,0,0,0,0,
// 0,0,0,0,0,0,0,0,
// 0,0,0,0,0,0,0,0,
// 0,0,0,0,0,0,0,0,
// };
// matrice della lettera "R" maiuscola, corsiva - matrix italic character "R"
int carR [65] = {
99, // il valore "99" e' un filler, usato per riempire il primo (inutile) campo della tabella
0,0,0,0,0,0,0,0,
1,1,1,0,0,0,0,0,
1,0,0,1,0,0,0,0,
1,0,0,1,0,0,0,0,
1,1,1,0,0,0,0,0,
1,1,0,0,0,0,0,0,
1,0,1,0,1,0,0,0,
1,0,0,1,0,0,0,0,
};
// matrice della lettera i minuscola, corsiva - matrix italic character "i"
int carI [65] = {
99, // il valore "99" e' un filler, usato per riempire il primo (inutile) campo della tabella
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,
0,0,1,0,0,0,0,0,
0,0,1,0,0,0,0,0,
0,0,1,0,0,0,0,0,
0,0,0,1,0,0,0,0,
};
// matrice della lettera m minuscola, corsiva - matrix italic character "m"
int carM [65] = {
99,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,1,1,0,1,1,0,0,
1,0,0,1,0,0,1,0,
1,0,0,1,0,0,1,0,
1,0,0,1,0,0,0,1,
};
// Matrice della lettera n minuscola, corsiva - matrix italic character "n"
int carN [65] = {
```

## Arduino: matrice led 8x8, scrivi e disegna - write and draw on 8x8 led matrix

```
99,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,1,1,0,0,0,0,
0,1,0,0,1,0,0,0,
0,1,0,0,1,0,0,0,
0,1,0,0,0,1,0,0,
};
// matrice per la definizione del primo disegno dell'animazione - first animation designs
int ca [65] = {
99,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,1,0,1,0,0,
0,0,0,1,1,1,0,0,
0,0,0,1,1,1,0,0,
0,0,0,1,0,0,0,
};
// matrice per la definizione del secondo disegno dell'animazione - second animation designs
int cb [65] = {
99,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,1,0,1,0,0,
0,0,1,1,1,1,0,0,
0,0,1,1,1,1,1,0,
0,0,0,1,1,1,0,0,
0,0,0,0,1,0,0,0,
};
// matrice per la definizione del quarto disegno dell'animazione - fourth animation designs
int cd [65] = {
99,
0,0,0,0,0,0,0,0,
0,0,1,1,0,1,1,0,
0,1,0,0,1,0,0,1,
0,1,0,0,0,0,0,1,
0,1,0,0,0,0,0,1,
0,0,1,0,0,0,1,0,
0,0,0,1,0,1,0,0,
0,0,0,0,1,0,0,0,
};
// matrice per la definizione del terzo disegno dell'animazione - third animation designs
int cc [65] = {
99,
0,0,0,0,0,0,0,0,
0,0,1,1,0,1,1,0,
0,1,1,1,1,1,1,1,
0,1,1,1,1,1,1,1,
0,1,1,1,1,1,1,1,
0,0,1,1,1,1,1,0,
0,0,0,1,1,1,0,0,
0,0,0,0,1,0,0,0,
};
// *****routine di esposizione della matrice memorizzata in tabella "comodo"*****
// ***** shows matrix stored in "comodo" table *****
void esponi (void)
{
for (refr = 1;refr<=300; refr++) // ripete 300 volte la routine di esposizione (fa il refresh) per
{ // ottenere qualche attimo di visibilita', prima di passare alla matrice successiva -
// refreshing cycle, 300 times before shows next matrix)
//
for (r1 = 1; r1 <= 8; r1 ++)
{
for (c1 = 1; c1 <= 8; c1 ++)
{
if (comodo [(r1-1)*8 + c1] ==1){ // se il byte in esame e' = 1 (led da accendere)
digitalWrite (rows [r1-1], HIGH); // attiva la riga (anodo)
digitalWrite (cols [c1-1], LOW); // disattiva la colonna (catodo) ed accende il led
sul display
}
// in ogni caso spegne il led in esame. Anche se e' stato appena acceso l'effetto
// persistenza lo manterra' comunque visibile fino al prossimo refresh:
digitalWrite (rows [r1-1], LOW); // disattiva la riga (anodo)
}
}
}
}
```

## Arduino: matrice led 8x8, scrivi e disegna - write and draw on 8x8 led matrix

```
        digitalWrite (cols [c1-1], HIGH);    // attiva la colonna (catodo) e spegne il led
    }
}
}
// *** routine di preparazione della colonna da inserire a destra del display in scorrimento ***
// *****scrolling routine - first right column prepration *****
void scorri (void)
{
    // ciclo di preparazione della nuova colonna da visualizzare
    for (c = 1; c <=8;c++)
    {
        for (r = 1; r<=8; r++)
        {
            colonna [r-1] = carcomodo [(r-1)*8 + c]; // trasferisce nella tabella "colonna" la colonna in
            // esame della matrice da esporre
            if (colonna[r-1] == 1) semaforo = 1; // attiva il semaforo di "colonna piena" se almeno un
            // led della colonna deve essere acceso
        }
        if (semaforo ==1)
        {
            // al cambio di colonna verifica se deve essere
            // visualizzata la colonna precedente
            visualizzacolonna (); // se deve essere visualizzata lancia la routine di gestione scorrimento
            semaforo = 0;
        }
    }
}
// ***** routine di preparazione della matrice da esporre in modalita' scorrimento ****
// ***** building matrix to scroll *****
void visualizzacolonna (void)
{
    for (c2 = 1; c2<=7; c2++)
    {
        for (r2 = 1; r2<=8; r2++)
        {
            comodo [(r2-1)*8+c2] = comodo [(r2-1)*8+c2+1]; // sposta a sinistra ogni colonna
            // della matrice "comodo"
            if (c2 == 7) comodo [(r2-1)*8+c2+1] = colonna [r2-1]; // inserisce la nuova colonna da
            // esporre nell'ottava colonna della matrice "comodo"
        }
    }
    esponi (); // lancia la routine di visualizzazione della matrice "comodo"
}
//
//
void setup()
{
    // definisce come output le porte definite nella tabella pins
    for (i = 1; i <= 16; i++)
    {
        pinMode(pins[i], OUTPUT);
    }
}
//
//
void loop()
{
    for (i=1; i<=8;i++)
    {
        // spegne preventivamente tutti i led del display
        digitalWrite(rows[i - 1], LOW); // spegne tutte le righe (anodi) e spegne tutti i led
        digitalWrite(cols[i - 1], LOW); // spegne tutte le colonne (catodi) e predispone la matrice
        // al suo utilizzo nel programma
    }
    // ciclo di avvio: accende e spegne progressivamente, riga per riga, tutti i led
    // starting cycle, turn on and off each led, row by row for (i = 1; i <= 8; i++)
    {
        digitalWrite(rows[i - 1], HIGH); // accende progressivamente tutte le righe
        delay(100);
    }
    for (i = 1; i <= 8; i++)
    {
        digitalWrite(rows[i - 1], LOW); // spegne progressivamente tutte le righe
        delay(100);
    }
    // ***** modalita' di scorrimento del testo*****
    // ***** scrolling text *****
    for (i=1;i<=8;i++)
    {
        digitalWrite (cols [i-1], HIGH); // mette in stato di HIGH tutte le colonne (i catodi)
        digitalWrite (rows [i-1], LOW); // mette in stato di LOW tutte le righe (gli anodi)
    }
}
```

## Arduino: matrice led 8x8, scrivi e disegna - write and draw on 8x8 led matrix

```
for (z = 1; z<=64; z++)
{
  carcomodo [z] = 0; // pulisce le tabelle (le matrici) di lavoro
  comodo [z] = 0;
}
// ***** gestione della lettera R - manages letter "R"
for (i=0; i<=64;i++)
{
  carcomodo [i] = carR [i]; // inserisce la matrice della lettera R in carcomodo, che e' la
// tabella di input alla routine di scorrimento
}
scorri (); // lancia la routine di scorrimento e visualizzazione della matrice
// ***** gestione della lettera i - manages letter "i"
for (i=0; i<=64;i++)
{
  carcomodo [i] = carI [i];
}

scorri ();
// ***** gestione della lettera m - manages letter "m"
for (i=0; i<=64;i++)
{
  carcomodo [i] = carM [i];
}
scorri ();
// ***** gestione della lettera i - manages letter "i"
for (i=0; i<=64;i++)
{
  carcomodo [i] = carI [i];
}
scorri ();
// ***** gestione della lettera n - manages letter "n"
for (i=0; i<=64;i++)
{
  carcomodo [i] = carN [i];
}
scorri ();
// ***** gestione della lettera i - manages letter "i"
for (i=0; i<=64;i++)
{
  carcomodo [i] = carI [i];
}
scorri ();
// ***** gestione della fase di fine scorrimento - scroll ending phase
for (i =0;i<=7;i++)
{
  colonna [i] = 0; // Prepara una colonna spenta (la tabella colonna e' usata
// dalla routine visualizzacolonna)
}
for (z=0;z<=8;z++)
{
  visualizzacolonna (); // visualizza la colonna spenta, facendola scorrere 8 volte sul display
}
// ***** animazione finale - final animation *****
//
// ***** visualizzazione del primo sprite - first sprite *****
for (z = 0; z <=64;z++)
{
  comodo [z] = ca [z]; // trasferisce nella tabella "comodo" la matrice da visualizzare
}
esponi();
// ***** visualizzazione del secondo sprite - second sprite *****
for (z = 0; z <=64;z++)
{
  comodo [z] = cb [z]; // trasferisce nella tabella "comodo" la matrice da visualizzare
}
esponi();
for (int a = 0; a<=4;a++)
{
  //ripete per cinque volte gli ultimi due sprite della sequenza animata
  // repeat five times last two sprites
  //
  // ***** routine di visualizzazione del terzo sprite - third sprite *****
  for (z = 0; z <=64;z++)
  {
    comodo [z] = cc [z]; // trasferisce nella tabella "comodo" la matrice da visualizzare
  }
}
```



## Arduino: matrice led 8x8, scrivi e disegna - write and draw on 8x8 led matrix

```
esponi();
// ***** routine di visualizzazione del quarto sprite - last sprite *****
for (z = 0; z <=64;z++)
{
  comodo [z] = cd [z]; // trasferisce nella tabella "comodo" la matrice da visualizzare
}
esponi();
}
}
```