**Arduino – lezione 08: display a 7 segmenti e creazione di librerie**

Pubblicato il [Febbraio 19, 2014](https://www.maffucci.it/2014/02/19/arduino-lezione-08-display-a-7-segmenti-e-creazione-di-librerie/) da [admin](https://www.maffucci.it/author/admin/%22%20%5Co%20%22Visualizza%20tutti%20gli%20articoli%20di%20admin)

Finalmente una nuova lezione

Inserisco una lezione sull’uso dei display a 7 segmenti che recentemente ho realizzato con una mia classe 5′, lo scopo è quello di comprendere meglio l’uso dei cicli e il conteggio ed incominciare a capire come creare le librerie.

Mi limiterò ad utilizzare un solo display è questo verrà utilizzato per contare da 0 a 9, dopo di che verrà utilizzato per visualizzare in sequenza solo i numeri pari e poi solo numeri dispari e poi cicli alternati di conteggi pari e dispari, ritengo un buon modo per allenarsi con condizioni e cicli.

Per ultimo illustrerò creare rapidamente una libreria da includere nel proprio codice.

Utilizzerò per questa lezione un display a 7 segmenti a catodo comune e la scelta non è dettata da motivi tecnici ma semplicemente perché nel magazzino della scuola ne abbiamo una scorta notevole

Innanzitutto vediamo di capire cosa è e come funziona un display a 7 segmenti.

Un display a 7 segmenti è tra i display più usati è realizzato da 7 segmenti ed ciascuno di essi è costituito da uno o più led disposti in modo da ottenere la disposizione indicata nelle figura che segue:



Il dispositivo è detto dual in line. I sette ingressi sono nominati con lelettere dell’alfabeto: a, b, c, d, e, f, g inoltre disponiamo di un’ilteriore ingresso nominato DP (Decimal Point).

Un display viene detto a catodo comune se i 7 catodi dei LED sono connessi insieme e i rispettivi anodi sono disponibili così come rappresentato nella figura che segue:



Un display viene detto ad anodo comune se i 7 anodi dei LED sono connessi insieme e i rispettivi catodi sono disponibili così come rappresentato nella figura che segue:



Per poter illuminare un segmento del display è necessario che il corrispondente LED venga attraversato da corrente, ovvero l’anodo deve risultare positivo rispetto al catodo. Ovviamente per limitare la quantità di corrente che attraversa ogni LED ad un valore tra i 10 e i 20 mA è indispensabile inserire una resistenza di limitazione.

Nel caso del display che dispongo, HDSP 5503, si ha per ogni singolo LED:



Inoltre sappiamo che la tensione di alimentazione è di 5V

Possiamo calcolare la resistenza da insierire in serie ad ogni LED:



Anche in questo caso non me ne vorrete se utilizzo una resistenza di valore un po’ più elevato, da [pmath size=12]180 Omega[/pmath] semplicemente perché ne dispongo di un gran numero, avrò una luminosità leggermente inferiore e sicuramente il display mi durerà un po’ di più.

**Collegamento del display**



**Collegamento reale**



**Schetch**

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133134135136137138139140141 | // Prof. Michele Maffucci// 19.02.2014 // Conteggio 9 secondi con display 7 segmenti a catodo comune // definizione delle variabili a cui collegare ogni segmento del display// la lettera rappresenta il segmento del display il numero il pin// digitale di Arduino a cui è collegato il corrispondente pin del display const int a=7;const int b=6;const int c=4;const int d=2;const int e=1;const int f=9;const int g=10; void LedAcceso(int n){  switch(n)  {  case 0:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, LOW);    break;   case 1:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, LOW);    break;   case 2:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, LOW);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    break;  case 3:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    break;   case 4:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 5:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 6:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 7:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, LOW);    break;   case 8:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 9:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;  }} void setup(){  pinMode(a, OUTPUT);   pinMode(b, OUTPUT);   pinMode(c, OUTPUT);   pinMode(d, OUTPUT);   pinMode(e, OUTPUT);   pinMode(f, OUTPUT);   pinMode(g, OUTPUT);} void loop (){  for (int i=0;i<10;i++)   {    LedAcceso(i);    delay (1000);  }} |

**Funzionamento**

La parte iniziale dello sketch definisce le variabile che utilizzeremo per poter accendere il corrispondente LED del display, come indicato nel codice le lettere rappresentano il segmento del display, i numeri i pin digitali di Arduino. Per evitare errori di collegamento ho preferito una corrispondenza di numero tra pin Arduino e pin display.

|  |  |
| --- | --- |
| 1234567891011 | // definizione delle variabili a cui collegare ogni segmento del display// la lettera rappresenta il segmento del display il numero il pin// digitale di Arduino a cui è collegato il corrispondente pin del display const int a=7;const int b=6;const int c=4;const int d=2;const int e=1;const int f=9;const int g=10; |

Nel setup vengono definiamo di outuput tutti i piedini utilizzati per accendere i vari segmenti:

|  |  |
| --- | --- |
| 12345678910 | void setup(){  pinMode(a, OUTPUT);   pinMode(b, OUTPUT);   pinMode(c, OUTPUT);   pinMode(d, OUTPUT);   pinMode(e, OUTPUT);   pinMode(f, OUTPUT);   pinMode(g, OUTPUT);} |

Il loop è costituito da un ciclo for che incrementa la variabile i di 1 unità alla volta (infatti abbiamo i++)

|  |  |
| --- | --- |
| 12345678 | void loop (){  for (int i=0;i<10;i++)   {    LedAcceso(i);    delay (1000);  }} |

Ad ogni ciclo viene chiamata la funzione:



|  |  |
| --- | --- |
| 1 | LedAcceso(i); |

che ha un unico parametro intero i, assegnato ad ogni ciclo dal for all’interno del loop.

La funzione LedAcceso(i) è costituita da uno switch che in funzione del valore n passato (ricordate che n è un intero che viene passato tramite il for incluso nel loop), non appena vi è la corrispondenza con uno dei case dello switch, vengono posti ad HIGH o a LOW i corrispondente segmenti del display, il break consente di uscire dallo switch e dalla funzione, il controllo ritorna al loop in cui troviamo un interruzione del loop di 1 secondo, ciò consente di mantenere accesi i led del display per 1 secondo. Il ciclo for nel loop prosegue fino a quando è vera la condizione i<10, quando la condizione non è più vera il ciclo incomincia nuovamente da 0.

|  |  |
| --- | --- |
| 123456789101112131415 | void LedAcceso(int n){  switch(n)  {  case 0:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, LOW);    break; ... |

Come secondo esercizio proviamo ora a realizzare uno sketch in grado di contare per due, quindi visualizzeremo sul display solamente i numeri 0, 2, 4, 6, 8. Ovviamente il modo più banale per risolvere il problema potrebbe essere quello di eliminare o commentare i corrispondenti case all’interno della funzione LedAcceso(). Operare in questo modo però non è ottimale, perché potrei avere la necessità, come accadrà più avanti, di visualizzare solamente i numeri dispari, oppure alternare conteggi di numeri pari e numeri dispari, quindi megli operare all’interno del loop.

Utilizziamo anche in questo caso un ciclo for per effettuare il conteggio da 0 a 9 e per, nel caso in cui il valore della variabile i corrisponde ad un numero pari, allora visualizzaimo su display il numero.

Vi ricordo che un numero è pari se il resto della divisione per 2 è uguale a 0, quindi solo quando il resto è 0 visualizziamo il numero sul display.

Questo lo sketch:

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133134135136137138139140141142143144145146147148149150151152 | // Prof. Michele Maffucci// 19.02.2014 // Conteggio numeri pari da 0 a 8 con display 7 segmenti a catodo comune // definizione delle variabili a cui collegare ogni segmento del display// la lettera rappresenta il segmento del display il numero il pin// digitale di Arduino a cui è collegato il corrispondente pin del display const int a=7;const int b=6;const int c=4;const int d=2;const int e=1;const int f=9;const int g=10; void AccendiLed(int n){  switch(n)  {  case 0:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, LOW);    break;  case 1:     digitalWrite(a, LOW);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, LOW);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, LOW);     break;   case 2:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, LOW);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    break; case 3:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, HIGH);     break;   case 4:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 5:     digitalWrite(a, HIGH);     digitalWrite(b, LOW);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, HIGH);     digitalWrite(g, HIGH);     break;   case 6:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 7:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, LOW);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, LOW);     break;   case 8:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 9:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, HIGH);     digitalWrite(g, HIGH);     break;  }} void setup(){  pinMode(a, OUTPUT);   pinMode(b, OUTPUT);   pinMode(c, OUTPUT);   pinMode(d, OUTPUT);   pinMode(e, OUTPUT);   pinMode(f, OUTPUT);   pinMode(g, OUTPUT);} void loop (){  int div=0;     // variabile, inizializzata a 0 in cui memorizzare                 // il risultato della divisione  int resto=0;   // variabile, inizializzata a 0 in cui memorizzare                 // il resto della divisione   for (int i=0;i<10;i++)  {    div=i/2;         // dividiamo il numero attuale per 2    resto=i-(div\*2); // calcoliamo il resto che sarà zero solo quando i è pari     if (resto==0) {  // se resto è uguale a 0 allora viene chiamata la                     // funzione AccendiLed() ciò accadrà solo per i pari      AccendiLed(i);      delay (1000);    }  }} |

In maniera più elegante possiamo effettuare il controllo se il numero è pari o dispari utilizzando la funzione modulo indicata con il simbolo % che consente di restituire il resto della divisione tra due numeri, in questo modo il codice all’interno del loop diventa più breve:

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133134135136137138139140141142143144145 | // Prof. Michele Maffucci// 19.02.2014 // Conteggio numeri pari da 0 a 8 con display 7 segmenti a catodo comune // definizione delle variabili a cui collegare ogni segmento del display// la lettera rappresenta il segmento del display il numero il pin// digitale di Arduino a cui è collegato il corrispondente pin del display const int a=7;const int b=6;const int c=4;const int d=2;const int e=1;const int f=9;const int g=10; void AccendiLed(int n){  switch(n)  {  case 0:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, LOW);    break;   case 1:     digitalWrite(a, LOW);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, LOW);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, LOW);     break;   case 2:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, LOW);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    break;  case 3:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, HIGH);     break;   case 4:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 5:     digitalWrite(a, HIGH);     digitalWrite(b, LOW);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, HIGH);     digitalWrite(g, HIGH);     break;   case 6:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 7:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, LOW);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, LOW);     break;   case 8:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 9:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, HIGH);     digitalWrite(g, HIGH);     break;  }} void setup(){  pinMode(a, OUTPUT);   pinMode(b, OUTPUT);   pinMode(c, OUTPUT);   pinMode(d, OUTPUT);   pinMode(e, OUTPUT);   pinMode(f, OUTPUT);   pinMode(g, OUTPUT);} void loop (){  for (int i=0;i<10;i++)   {    if (i%2==0) {     // se il resto della divisione è uguale a 0                      // allora viene viene chiamata la funzione AccendiLed()                      // che visualizzerà il corrispondente numero pari      AccendiLed(i);      delay (1000);    }  }} |

Nel caso in cui si desideri visualizzare solamente i numeri dispari, sarà sufficiente controllare quando il resto della divisione NON è uguale a zero, pertanto lo schetch diventa il seguente:

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133134135136137138139140141142143144145 | // Prof. Michele Maffucci// 19.02.2014 // Conteggio numeri dispari da 1 a 9 con display 7 segmenti a catodo comune // definizione delle variabili a cui collegare ogni segmento del display// la lettera rappresenta il segmento del display il numero il pin// digitale di Arduino a cui è collegato il corrispondente pin del display const int a=7;const int b=6;const int c=4;const int d=2;const int e=1;const int f=9;const int g=10; void AccendiLed(int n){  switch(n)  {  case 0:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, LOW);    break;   case 1:     digitalWrite(a, LOW);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, LOW);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, LOW);     break;   case 2:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, LOW);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    break;  case 3:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, HIGH);     break;   case 4:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 5:     digitalWrite(a, HIGH);     digitalWrite(b, LOW);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, HIGH);     digitalWrite(g, HIGH);     break;   case 6:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 7:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, LOW);      digitalWrite(e, LOW);     digitalWrite(f, LOW);     digitalWrite(g, LOW);     break;   case 8:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    break;   case 9:     digitalWrite(a, HIGH);     digitalWrite(b, HIGH);     digitalWrite(c, HIGH);      digitalWrite(d, HIGH);      digitalWrite(e, LOW);     digitalWrite(f, HIGH);     digitalWrite(g, HIGH);     break;  }} void setup(){  pinMode(a, OUTPUT);   pinMode(b, OUTPUT);   pinMode(c, OUTPUT);   pinMode(d, OUTPUT);   pinMode(e, OUTPUT);   pinMode(f, OUTPUT);   pinMode(g, OUTPUT);} void loop (){  for (int i=0;i<10;i++)   {    if (i%2!=0) {     // se il resto della divisione è diverso da 0                      // allora viene viene chiamata la funzione AccendiLed()                      // che visualizzerà il corrispondente numero dispari      AccendiLed(i);      delay (1000);    }  }} |

Realizziamo ora uno sketch che permette di visualizzare prima un ciclo di numeri pari e poi un ciclo di numeri dispari, in questo caso sarà sufficiente accodare due cicli for all’interno del loop, il primo che realizza il conteggio dei numeri pari ed il secondo che realizza il conteggio dei numeri dispari, vediamo come:

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133134135136137138139140141142143144145146147148149150151152153154155156157158159160161162163164165166167168169170171172173174175176177178 | // Prof. Michele Maffucci// 19.02.2014 // Conteggio cicli pari e dispari per gli interi da 0 a 9 con display 7 segmenti a catodo comune // definizione delle variabili a cui collegare ogni segmento del display// la lettera rappresenta il segmento del display il numero il pin// digitale di Arduino a cui è collegato il corrispondente pin del display const int a=7;const int b=6;const int c=4;const int d=2;const int e=1;const int f=9;const int g=10;const int h=8; void AccendiLed(int n){  switch(n)  {  case 0:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, LOW);    digitalWrite(h, LOW);    break;   case 1:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, LOW);    digitalWrite(h, LOW);    break;   case 2:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, LOW);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;  case 3:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 4:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 5:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 6:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 7:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, LOW);    digitalWrite(h, LOW);    break;   case 8:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 9:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;  }} void setup(){  pinMode(a, OUTPUT);   pinMode(b, OUTPUT);   pinMode(c, OUTPUT);   pinMode(d, OUTPUT);   pinMode(e, OUTPUT);   pinMode(f, OUTPUT);   pinMode(g, OUTPUT);  pinMode(h, OUTPUT);} void loop (){   // ciclo pari   for (int i=0;i<10;i++) // ciclo che incrementa il numero da visualizzare  {     // nel caso in cui il resto della division è 0,    // non abbiamo raggiunto il valore massimo    // visualizziamo solamente i numeri pari     if (i%2==0 && i!=10) {      AccendiLed(i);      delay (1000);    }  }   // ciclo dispari   for (int i=0;i<10;i++)   {     // nel caso in cui il resto della division è diverso da 0,    // non abbiamo raggiunto il valore massimo    // visualizziamo solamente i numeri dispari     if (i%2!=0 && i!=10) {      AccendiLed(i);      delay (1000);    }  } } |

Poiché la modalità con cui viene controllato il display a 7 segmenti è sempre la stesso potrebbe essere opportuno definire una libreria generica a cui riferirsi quando vogliamo usare un display come quello impiegato in questa lezione, vediamo come.
Utilizziamo come sketch principale il conteggio alternato pari dispari appena realizzato.
Bisogna che la funzione AccendiLed() diventi la libreria da includere nello sketch precedente.

Create nella cartella libraries di arduino la cartella “AccendiLed”:



ora con il vostro editor di testi preferito create il file AccendiLed.cpp in cui andremo ad inserire la funzione AccendiLed() presente nello sketch precedente, salvate questo file nella cartella appena creata libraries -> AccendiLed

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133 | /\* AccendiLed.cpplibreria che consente il controllo di un display a 7 segmenti a catodo comunedel tipo HDSP 5503\*/ #include "Arduino.h"#include "AccendiLed.h" // costanti che definiscono i pin del display const int a=7;const int b=6;const int c=4;const int d=2;const int e=1;const int f=9;const int g=10;const int h=8; void AccendiLed(int n){  switch(n)  {  case 0:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, LOW);    digitalWrite(h, LOW);    break;   case 1:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, LOW);    digitalWrite(h, LOW);    break;   case 2:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, LOW);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;  case 3:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 4:    digitalWrite(a, LOW);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 5:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 6:    digitalWrite(a, HIGH);    digitalWrite(b, LOW);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 7:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, LOW);     digitalWrite(e, LOW);    digitalWrite(f, LOW);    digitalWrite(g, LOW);    digitalWrite(h, LOW);    break;   case 8:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, HIGH);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;   case 9:    digitalWrite(a, HIGH);    digitalWrite(b, HIGH);    digitalWrite(c, HIGH);     digitalWrite(d, HIGH);     digitalWrite(e, LOW);    digitalWrite(f, HIGH);    digitalWrite(g, HIGH);    digitalWrite(h, LOW);    break;  }} |

Nella libreria adesso vi saranno i due file .cpp e .h



Il file AccendiLed.cpp contiene il codice per il controllo del display ma anche le seguenti due linee di codice:

|  |  |
| --- | --- |
| 12 | #include "Arduino.h"#include "AccendiLed.h" |

La riga #include “Arduino.h” è necessaria per far si che la libreria che stiamo creando possa utilizzare il set di istruzioni di Arduino (funzioni e costanti). Senza queste due righe il compilatore restituirebbe errore per tutte le funzioni di Arduino che vengono utilizzate nel nostro sketch.

La riga #include “AccendiLed.h” contiene la definizione delle funzioni, in altro modo detti i prototipi delle funzioni.

Quando uno sketch viene compilato, vengono creati automaticamente tutti i prototipi per le funzioni utilizzate nello sketch, ma non i prototipi del codice delle librerie, quindi se costruiamo una libreria è necessario creare un header con questi prototipi.
Questo è l’header file che viene aggiunto all’inizio di uno sketch quando si importa una libreria dall’IDE.

Bisogna ora creare l’header file AccendiLed.h e salvarlo sempre in libraries -> AccendiLed

|  |  |
| --- | --- |
| 1234567 | /\* AccendiLed.hheader file della libreria AccendiLed\*/ #include "Arduino.h" void AccendiLed(int n);  // prototipo della funzione |

Tenete in conto che che l’IDE di Arduino aggiorna l’elenco delle librerie disponibili solo quando si riavvia l’IDE, quindi una volta creata la libreria chiudete e riavviate l’IDE.

Il codice dello sketch che visualizza la sequenza pari e dispari diventa:

|  |  |
| --- | --- |
| 12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667 | // Prof. Michele Maffucci// 19.02.2014 // Conteggio cicli pari e dispari per gli interi da 0 a 9 con display 7 segmenti a catodo comune // definizione delle variabili a cui collegare ogni segmento del display// la lettera rappresenta il segmento del display il numero il pin// digitale di Arduino a cui è collegato il corrispondente pin del display #include "AccendiLed.h" // costanti che definiscono i pin di Arduino a cui collegare il display const int a=7;const int b=6;const int c=4;const int d=2;const int e=1;const int f=9;const int g=10;const int h=8; void setup(){  pinMode(a, OUTPUT);   pinMode(b, OUTPUT);   pinMode(c, OUTPUT);   pinMode(d, OUTPUT);   pinMode(e, OUTPUT);   pinMode(f, OUTPUT);   pinMode(g, OUTPUT);  pinMode(h, OUTPUT);} void loop (){   // ciclo pari   for (int i=0;i<10;i++) // ciclo che incrementa il numero da visualizzare  {     // nel caso in cui il resto della division è 0,    // non abbiamo raggiunto il valore massimo    // visualizziamo solamente i numeri pari     if (i%2==0 && i!=10) {      AccendiLed(i);      delay (1000);    }  }   // ciclo dispari   for (int i=0;i<10;i++)   {     // nel caso in cui il resto della division è diverso da 0,    // non abbiamo raggiunto il valore massimo    // visualizziamo solamente i numeri dispari     if (i%2!=0 && i!=10) {      AccendiLed(i);      delay (1000);    }  } } |