

Utilizzare e Creare una Libreria per il Display a 7 Segmenti

Obiettivo: Utilizzare e creare una libreria (file header e cpp) per un Display a Sette Segmenti.

Puoi scaricare i file di libreria cliccando nel seguente link:
<http://www.arduinofacile.it/wp-content/uploads/2021/03/SevenSegment.zip>

I file scaricati devono essere inseriti all'interno della cartella di progetto insieme al file .ino

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Display a 7 Segmenti
- 8 Resistenze

Pre-requisiti:

[Creare funzioni con Arduino ... per un Display a 7 Segmenti](#)

Teoria: la realizzazione di funzioni di libreria permette di facilitare l'operazione di **riutilizzo del codice** rendendo più veloce e più rapido lo sviluppo. Nel caso specifico la funzione di libreria implementata sarà costituita da un file header (.h) e da un file sorgente (.cpp).

Un file header è un file di testo che contiene i prototipi dei metodi (funzioni) definite nel relativo file sorgente. Nel caso in questione il file header contiene anche la dichiarazione della classe "SevenSegment" utilizzata per modellare il display a sette segmenti.

Tale classe sarà caratterizzata da 10 attributi:

- **int pinA:** il pin A del display a sette segmenti
- **int pinB:** il pin B del display a sette segmenti
- **int pinC:** il pin C del display a sette segmenti
- **int pinD:** il pin D del display a sette segmenti
- **int pinE:** il pin E del display a sette segmenti
- **int pinF:** il pin F del display a sette segmenti
- **int pinG:** il pin G del display a sette segmenti
- **int pinDP:** il pin DP del display a sette segmenti
- **bool isCommonAnode:** indica se il display è di tipo anodo comune oppure no

e da 11 metodi

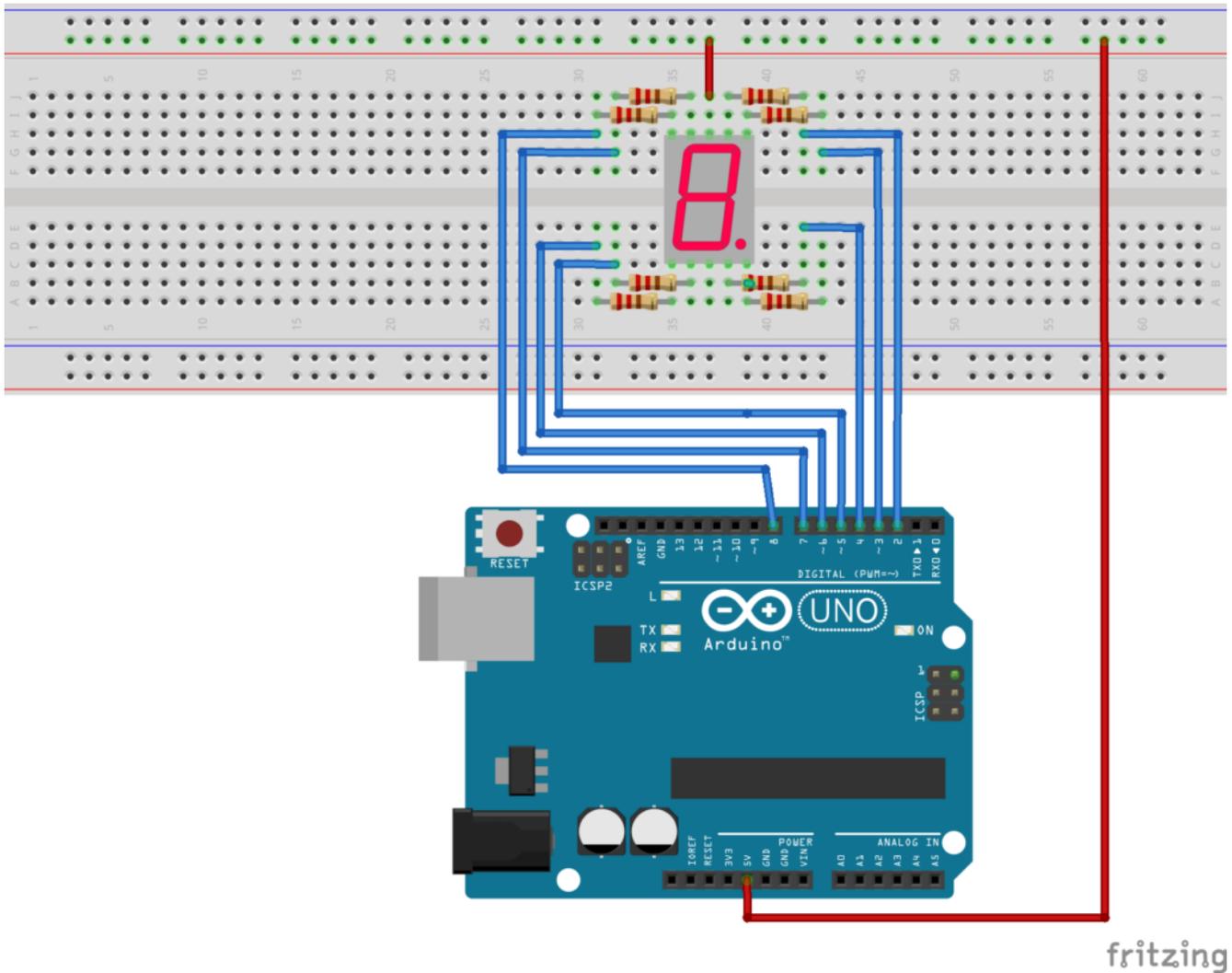
- **void Print0():** metodo utilizzato per stampare il numero 0
- **void Print1():** metodo utilizzato per stampare il numero 1
- **void Print2():** metodo utilizzato per stampare il numero 2
- **void Print3():** metodo utilizzato per stampare il numero

3

- **void Print4():** metodo utilizzato per stampare il numero 4
- **void Print5():** metodo utilizzato per stampare il numero 5
- **void Print6():** metodo utilizzato per stampare il numero 6
- **void Print7():** metodo utilizzato per stampare il numero 7
- **void Print8():** metodo utilizzato per stampare il numero 8
- **void Print9():** metodo utilizzato per stampare il numero 9
- **void Countdown():** metodo utilizzato per eseguire il countdown.

Nel file sorgente viene invece riportata l'implementazione dei prototipi delle funzioni dichiarate nel file header.

Collegamento Circuitale:



Collegamento Circuitale

Codice:

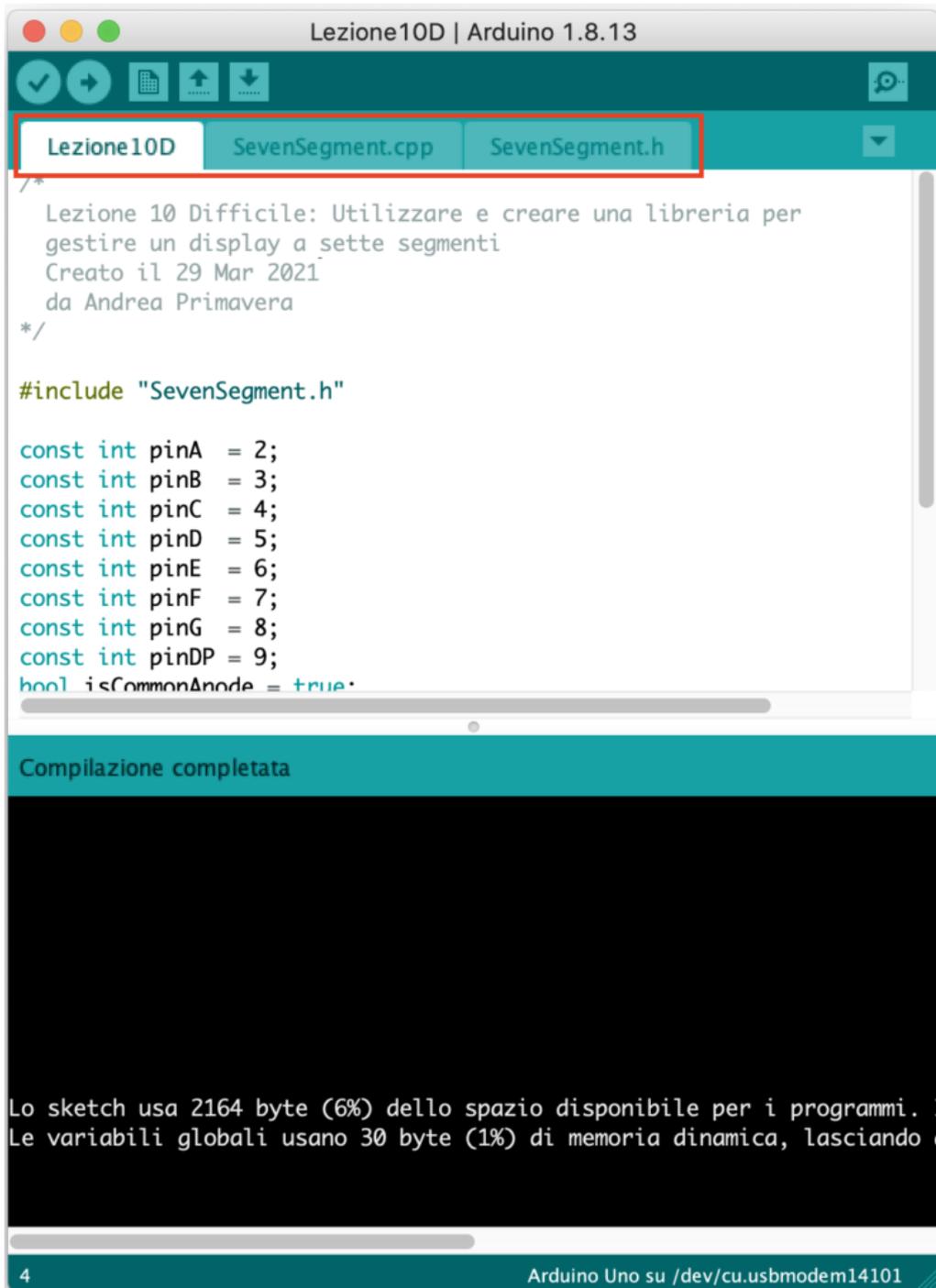
Vengono in seguito riportate le tre porzioni di codice utilizzate per creare la funzione di libreria relativa al display a sette segmenti.

- File Header: contiene la definizione della classe con i propri attributi (i.e., pinA, pinB, etc) ed i prototipi dei relativi metodi.

- File Sorgente: contiene le implementazioni dei metodi riportati nel file header.

- File Arduino: Utilizzato per fornire un esempio di come utilizzare la libreria per la gestione del display a sette segmenti.

Se tutti i file sono correttamente posizionati sullo stesso livello all'interno della cartella di progetto, due nuove tab compariranno nell'ambiente di sviluppo utilizzato per programmare Arduino. Attraverso queste tab sarà possibile visionare e modificare il file sorgente (.cpp) ed il file header (.h)



IDE con l'utilizzo della libreria SevenSegment.h

Utilizzare e Creare una

Libreria per il Sensore ad Ultrasuoni

Obiettivo: Utilizzare e creare una libreria (file header e cpp) per un Sensore a Ultrasuoni (HC-SR04) utilizzato per misurare la distanza.

Puoi scaricare i file di libreria cliccando nel seguente link:
<http://www.arduino facile.it/wp-content/uploads/2020/10/UltrasonicSensor.zip>

I file scaricati devono essere inseriti all'interno della cartella di progetto insieme al file .ino

Componenti elettronici:

- Arduino UNO
- Breadboard
- 1 Sensore Ultrasuoni (e.g., HC-SR04)

Pre-requisiti:

Il Sensore a Ultrasuoni

Teoria: la realizzazione di funzioni di libreria permette di facilitare l'operazione di **riutilizzo del codice** rendendo più veloce e più rapido lo sviluppo. Nel caso specifico la

funzione di libreria implementata sarà costituita da un file header (.h) e da un file sorgente (.cpp).

Un file header è un file di testo che contiene i prototipi dei metodi (funzioni) definite nel relativo file sorgente. Nel caso in questione il file header contiene anche la dichiarazione della classe "UltrasonicSensor" utilizzata per modellare il sensore ad ultrasuoni.

Tale classe sarà caratterizzata da 2 attributi:

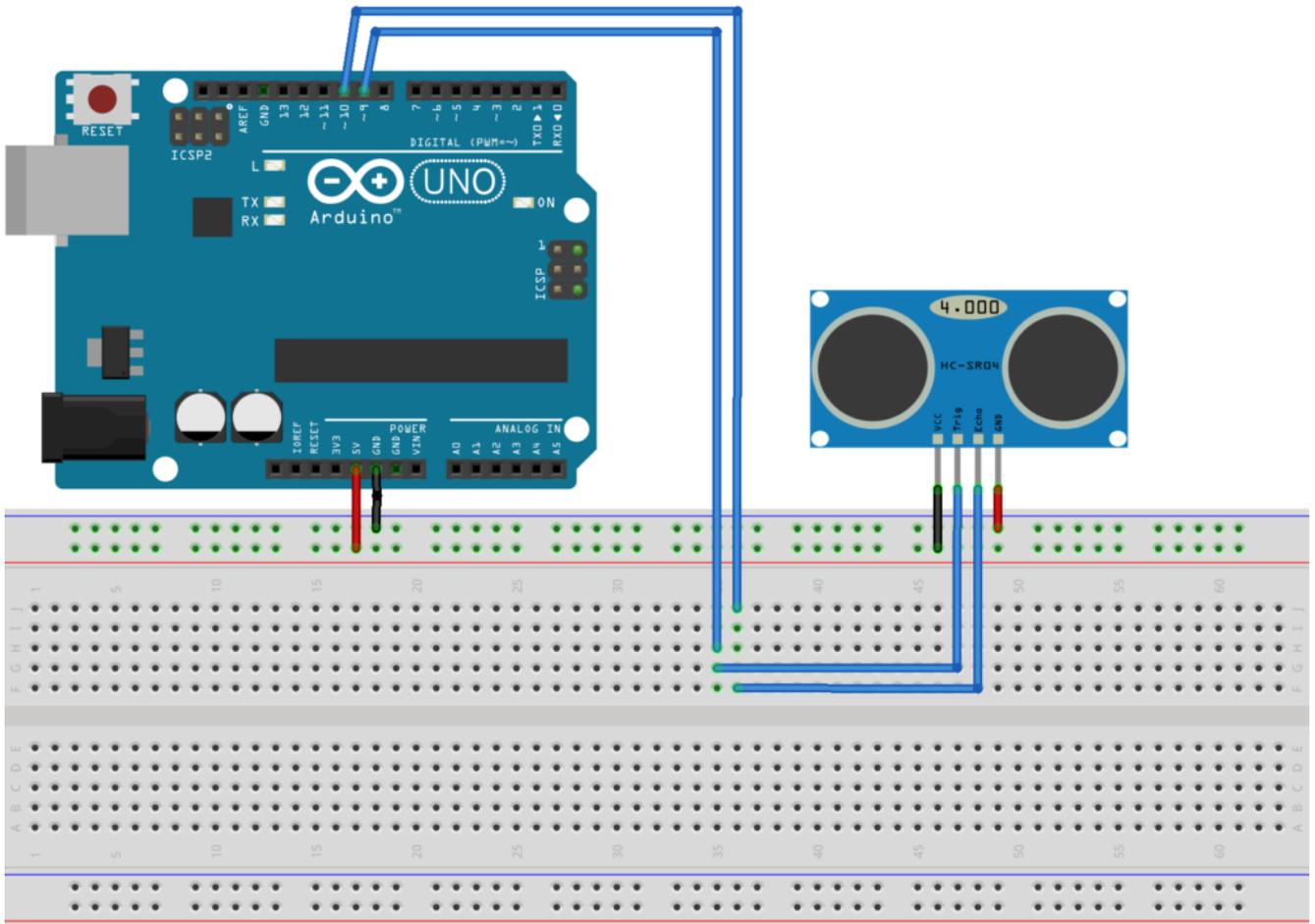
- **int pinEcho**: il pin di echo
- **int pinTrigger**: il pin di trigger

e da 4 metodi:

- **void SetPinEcho (int pinEcho)**: metodo utilizzato per settare il pin di echo
- **void SetPinTrigger(int pinTrigger)**: metodo utilizzato per settare il pin di trigger:
- **long GetDistance()**: metodo utilizzato per effettuare la misura di distanza (restituisce un valore di tipo long)
- **long GetAverageDistance(int numIteration)**: metodo utilizzato per effettuare la misura di distanza media su un numero dato di misure (restituisce un valore di tipo long)

Nel file sorgente viene invece riportata l'implementazione dei prototipi delle funzioni dichiarate nel file header.

Collegamento Circuitale:



fritzing

Collegamento Circuitale

Codice:

Vengono in seguito riportate le tre porzioni di codice utilizzate per creare la funzione di libreria relativa al sensore ad ultrasuoni HC-SR04.

- File Header: contiene la definizione della classe con i propri attributi (i.e., pinEcho e pinTrigger) ed i prototipi dei relativi metodi.

- File Sorgente: contiene le implementazioni dei metodi riportati nel file header.

- File Arduino: Utilizzato per fornire un esempio di come utilizzare la libreria per la gestione del sensore ad ultrasuoni.

Se tutti i file sono correttamente posizionati sullo stesso livello all'interno della cartella di progetto, due nuove tab compariranno nell'ambiente di sviluppo utilizzato per programmare Arduino. Attraverso queste tab sarà possibile visionare e modificare il file sorgente (.cpp) ed il file header (.h)

Lezione8D

UltrasonicSensor.cpp

UltrasonicSensor.h

```
/*  
  Lezione 8 Difficile: Utilizzare e creare una libreria per  
  il sensore ad Ultrasuoni  
  Creato il 13 Oct 2020  
  da Andrea Primavera  
*/  
  
#include "UltrasonicSensor.h"  
  
const int pinTrigger = 9;  
const int pinEcho    = 10;  
  
UltrasonicSensor US(pinEcho, pinTrigger);  
  
void setup() {  
  Serial.begin(9600);  
  Serial.print( "Sensore Ultrasuoni: ");  
}
```

ropbox/Arduino/ArduinoFacile/Difficile/Lezione8D/Lezione8D/Lezione8D.ino