

```
byte segDP = 6; // Quinto PIN del display 7Segment
byte segC = 7; // Quarto PIN del display 7Segment
byte segD = 8; // Secondo PIN del display 7Segment
byte segE = 9; // Primo PIN del display 7Segment
byte segG = 4; // Sesto PIN del display 7Segment
byte segF = 3; // Settimo PIN del display 7Segment
byte segA = 5; // Nono PIN del display 7Segment
byte segB = 2; // Decimo PIN del display 7Segment
byte pushButton = 11; // Il pulsante e' connesso al pin digitale 3
byte conta=0; // Contatore
const bool commonAnode= false; // False se è a Catodo comune (-)
static const byte pinArray[] = {segA, segB, segC, segD, segE, segF, segG };
// Le sequenze sottostanti indicano quali segmenti devono essere illuminati
static const byte HexCodeMap[] =
{
    // mappa segmenti ABCDEFG
    B00111111, // 0 "0" AAA
    B00000110, // 1 "1" F B
    B01011011, // 2 "2" F B
    B01001111, // 3 "3" GGG
    B01100110, // 4 "4" E C
    B01101101, // 5 "5" E C
    B01111101, // 6 "6" DDD
    B00000111, // 7 "7"
    B01111111, // 8 "8"
    B01101111, // 9 "9"
    B01110111, // 65 'A'
    B01111100, // 66 'b'
    B00111001, // 67 'C'
    B01011110, // 68 'd'
    B01111001, // 69 'E'
    B01110001 // 70 'F'
```

```
};

// Funzione che mostra la cifra "number" sul display

void ScriveCifra(byte number){

byte dataTmp = HexCodeMap[number];

for (byte c = 0; c < 7; c++, dataTmp = dataTmp >> 1){

digitalWrite(pinArray[c], commonAnode ? !(dataTmp & 0x01) : dataTmp & 0x01);

}

}

void setup(){

for (byte set=0; set<7; set++){

pinMode(pinArray[set],OUTPUT);

pinMode(segDP, OUTPUT);

pinMode(pushButton, INPUT);

ScriveCifra(conta); // mostro lo 0 (valore iniziale)

}

void loop(){

static int prebuttonState=-1; // Variabile statica per memorizzare il precedente stato

// Legge lo stato sul PIN 2

int buttonState = digitalRead(pushButton);

// Se premo il pulsante aumento il contatore e lo visualizzo

if ((buttonState!=prebuttonState) && (buttonState==1)){

conta++;

if (conta == 16) conta=0;

ScriveCifra(conta);

}

prebuttonState=buttonState;

delay(100); //tempo tra ogni lettura

}
```