

Programmi da scrivere per la lezione 7-8

### Sketch numero 1

salvare il seguente programma con il nome: 26\_74Hc595LedLcdFtc

```
/*
 * 26_74Hc595LedLcdFtc
 * utilizzo la libreria McMajan per gestire gli integrati
 * a cui ho connesso 8 led e un lcd
 * per visualizzare in forma scritta e visuale la luminosità rilevata da
 * una fotocellula connessa al pin A0
 */

#include <Ss_McMajan_Config.h>

// creo l'oggetto My595 per gestire i 2 integrati
hc595 My595(11,12,8,2); // latch12,clock11,data14,numero di 74hc595

int ftc = 0;
int luce = 0;
unsigned char leds;

void setup(){
  leds=0;
  // definisco il tipo di lcd usato
  My595.DisplayReset(LCD595_BASIC_DISPLAY_INIT | LCD595_MORELINES ,1); // display multilinea
  My595.SetCursor(0,0,1,1); // cursore a 0,0, display tipo 1 sul secondo 595
  My595.DisplayWrite("Setup is OK...",1); // stringa da scrivere sul secondo 595
  delay(3000); // attesa...
  My595.DisplayClean(1); // cancello display
}

void loop(){
  int leggiFtc = analogRead(ftc); // essendo un pin analogico non devo dichiarare nulla nel setup
  int numLedAccesi = leggiFtc / 101; // max valore ftc= 910 / 9 combinazioni = 101
  if (numLedAccesi > 8) numLedAccesi = 8;
  leds=0;
  for (int i = 0; i< numLedAccesi; i++) { // per ogni led acceso
    bitSet(leds,i); // metto a uno il bit corrispondente in leds
  }
  luce= map(leggiFtc,0,910,0,100); // trasformo in valore da 0 a 100 la lettura analogica
  regWrite();
  delay(500);
}

void regWrite(){
  My595.Set595Pin(leds,0); // accendo i led
  char myBuf[12]; // creo una variabile temporanea
  My595.DisplayClean(1); // pulisco il display collegato al chip 1
  My595.SetCursor(1,0,1,1); // posizione cursore carattere 2, riga 1, tipo 1, chip 1
  My595.DisplayWrite("Luminosita'= ",1); // scrivo sul chip 1
  My595.SetCursor(6,1,1,1); // posiziono cursore carattere 6, riga 2, tipo 1, chip 1
  My595.DisplayWrite(itoa(luce,myBuf,10), 1); // scrivo il numero contenuto in luce, dopo averlo trasformato in strings
  //My595.DisplayWrite(luce,1); // così non funziona!!
  My595.SetCursor(9,1,1,1); // mi posiziono dopo il numero appena scritto
  My595.DisplayWrite("%",1); // aggiungo il carattere percento
  My595.Send595(); // invio tutto quello che ho fatto prima alle uscite dei 74HC595
}
```

}

## Sketch numero 2

salvare il seguente programma con il nome: 27\_74Hc595LedLcdFtcStepper

```
[code]
/*
 * _74Hc595LedLcdFtcStepper
 * utilizzo la libreria McMajan per gestire gli integrati
 * a cui ho connesso 8 led e un lcd
 * per visualizzare in forma scritta e visuale la luminosità rilevata da
 * una fotocellula connessa al pin A0 e uno stepper motor ai pin 4,5,6,7
 */

#include <Ss_McMajan_Config.h>
#include <Stepper.h>

const int passiPerGiro = 2048;
int velocita = 10;
bool suGiu = 0;

//inizializzo lo stepper collegato da 4 a 7
Stepper myStepper(passiPerGiro, 4, 5, 6, 7);

// creo l'oggetto My595 per gestire i 2 integrati
hc595 My595(11,12,8,2); // latch12,clock11,data14,numero di 74hc595

int ftc = 0;
int luce = 0;
unsigned char leds;

void setup(){
  leds=0;
  // definisco il tipo di lcd usato
  My595.DisplayReset(LCD595_BASIC_DISPLAY_INIT | LCD595_MORELINES ,1); // display multilinea
  My595.SetCursor(0,0,1,1); // cursore a 0,0, display tipo 1 sul secondo 595
  My595.DisplayWrite("Setup is OK...",1); // stringa da scrivere sul secondo 595
  delay(3000); // attesa...
  My595.DisplayClean(1); // cancello display
}
void loop(){
  int leggiFtc = analogRead(ftc);
  int numLedAccesi = leggiFtc / 101; // max valore ftc= 910 / 9 combinazioni = 101
  if (numLedAccesi > 8) numLedAccesi = 8;
  leds=0;
  for (int i = 0; i< numLedAccesi; i++) { // per ogni led acceso
    bitSet(leds,i); // metto a uno il bit corrispondente in leds
  }
  luce= map(leggiFtc,0,910,0,100); // trasformo in valore da 0 a 100 la lettura analogica
  regWrite();

  // se il valore di luce è > 60% allora faccio fare 1 giro in senso orario al motore
  // 10 cicli da 21 passi per accelerare poi il resto
  if (luce > 60 && suGiu == 0) { // se suGiu = 0 allora la tenda è alzata
    for (velocita = 10; velocita <= 20; velocita++){ // posso abbassarla
      myStepper.setSpeed(velocita);
      myStepper.step(passiPerGiro/100);
      suGiu = 1; // suGiu = 1 tenda abbassata
    }
    myStepper.step(passiPerGiro-210);
  }
}
```

```

}
else if (luce < 61 && suGiu == 1){
    for (velocita = 10; velocita <= 20; velocita++){
        myStepper.setSpeed(velocita);
        myStepper.step(-passiPerGiro/100);
    }
    myStepper.step(-(passiPerGiro-210));
    suGiu = 0;
}
// metto tutti i pin usati dallo stepper a 1
for (int i =4 ; i<=7; i++) digitalWrite(i, LOW);
delay(500);
}

void regWrite(){
    My595.Set595Pin(leds,0);
    char myBuf[12];
    My595.DisplayClean(1);
    My595.SetCursor(1,0,1,1);
    My595.DisplayWrite("Luminosita'=",1);
    My595.SetCursor(6,1,1,1);
    My595.DisplayWrite(itoa(luce,myBuf,10), 1); // scrivo il numero contenuto in luce, dopo averlo trasformato in strings
    //My595.DisplayWrite(luce,1);
    My595.SetCursor(9,1,1,1);
    My595.DisplayWrite("%",1);
    My595.Send595();
}

```

// se suGiu = 1 tenda abbassata  
// posso alzarla  
// suGiu = 0 tenda alzata  
// in questo modo il motore dovrebbe scaldare di meno  
// accendo i led  
// creo una variabile temporanea  
// pulisco il display collegato al chip 1  
// posizione cursore carattere 2, riga 1, tipo 1, chip 1  
// scrivo sul chip 1  
// posiziono cursore carattere 6, riga 2, tipo 1, chip 1  
// così non funziona!!  
// mi posiziono dopo il numero appena scritto  
// aggiungo il carattere percento  
// invio tutto quello che ho fatto prima alle uscite dei 74HC595

---





