Progetti reali con ARDUINO

Introduzione alla scheda Arduino (parte 5ª)

gennaio 2015 – Giorgio Carpignano

I.I.S. PRIMO LEVI

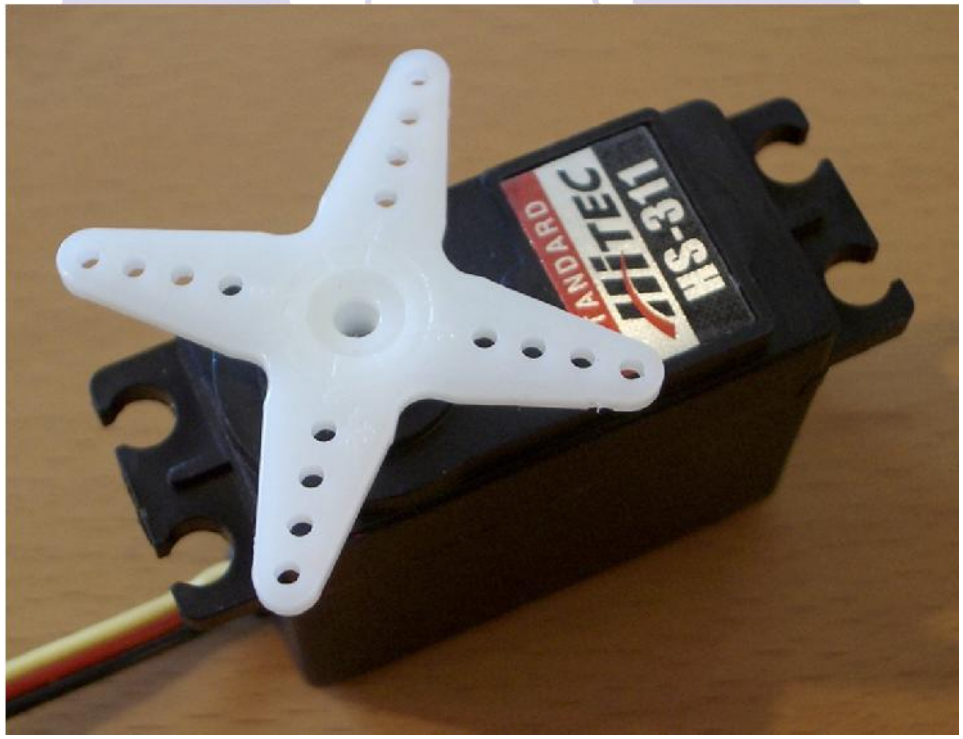
C.so Unione Sovietica 490 (TO)

Materiale didattico:

www.istitutoprimolevi.gov.it

Servomotori per radiocomandi

- Può essere posizionato con una rotazione dell'albero compreso tra **0°** e **180º**
- Un circuito di feedback interno (controreazione) e ingranaggi si prende cura di mantenere la posizione corretta
- Facile da utilizzare e collegare: utilizza solo 3 fili di collegamento con alimentazione a 5V.



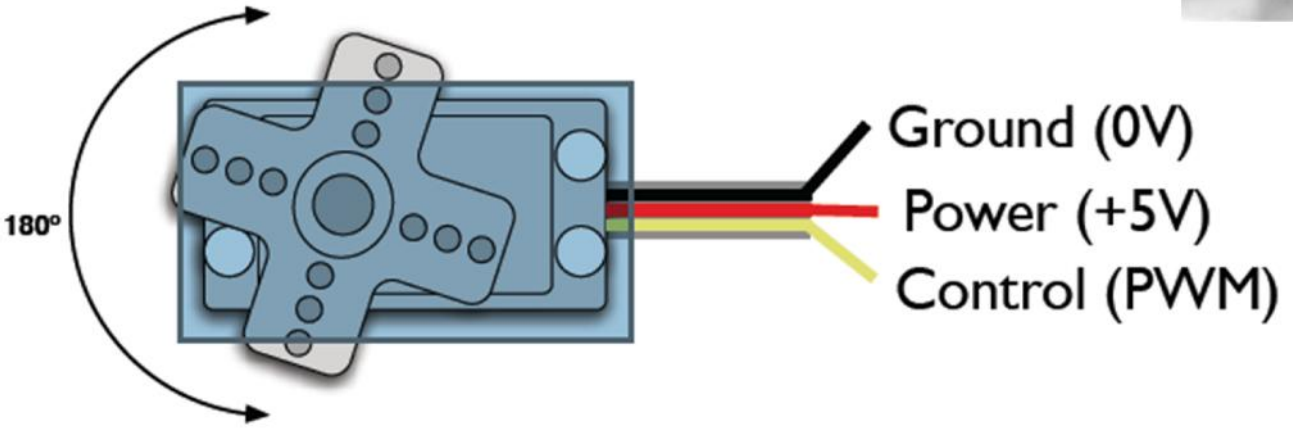
- In generale, il "servomotore" è un motore con un meccanismo di **feedback** (composto da sensore di posizionamento e sua logica di controllo) che permette di inviare comandi di posizione allo stesso senza che sia necessario effettuare la lettura di posizione per verificare la corretta posizione.

I servomotori dove si utilizzano?

- In robotica, negli effetti cinematografici, nei presepi e nei teatrini dei burattini si usano estensivamente.
- Ogni volta che avete bisogno di effettuare un piccolo controllo, un movimento ripetibile più volte.
- La rotazione dell'alberino può essere trasformata in un movimento lineare con un semplice circuito meccanico.

I servomotori

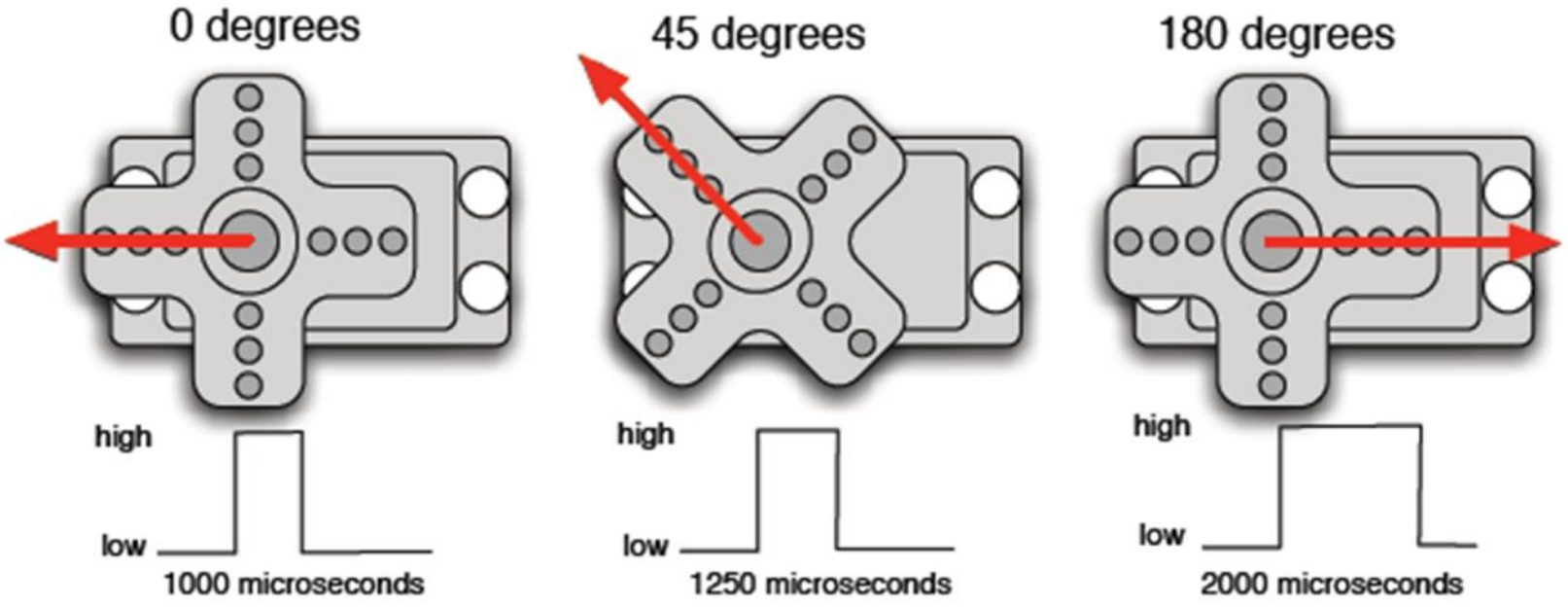
- Sono disponibili in molte dimensioni
- Dai modelli più piccoli (9 grammi)
- A quelli più grandi per le auto (157 grammi)
- TUTTI però possiedono solo 3 cavi di collegamento.
- La frequenza di aggiornamento del PWM è di 50 Hz (ogni 20 msec.)
- L'impulso varia da 1 a 2 msec.
- **1 msec** = posizione dell'alberino completamente ruotata con senso antiorario
- **2 msec** = posizione dell'alberino completamente ruotata con senso orario



Movimento dei servomotori

- Per posizionare il servomotore occorre trasmettere una serie di impulsi della durata da 1 a 2 msec.
- Per mantenere la posizione occorre ripetere periodicamente la trasmissione dell'impulso.
- E' necessario un certo tempo per ruotare. Se gli impulsi sono troppo veloci l'alberino non si sposta.

Ricorda → 1 msec. = 1000 «sec.

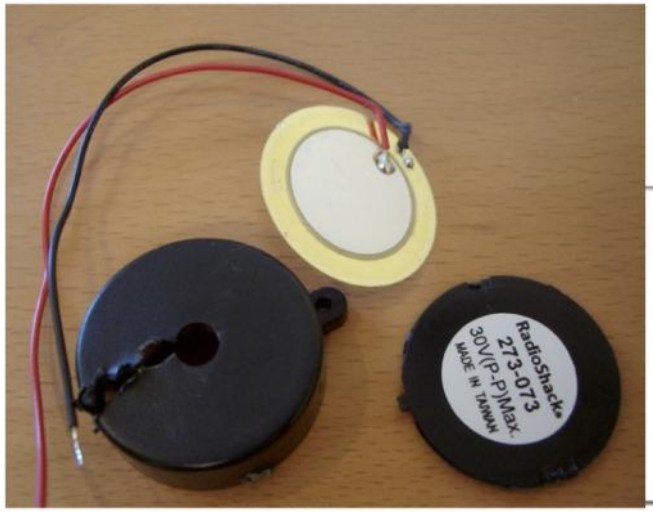


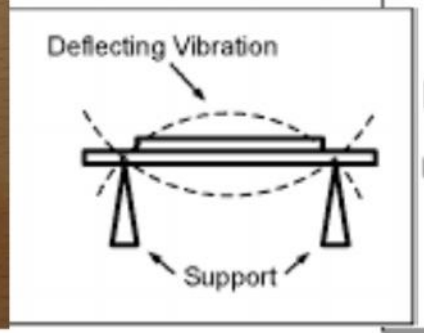
Buzzer piezoelettrico

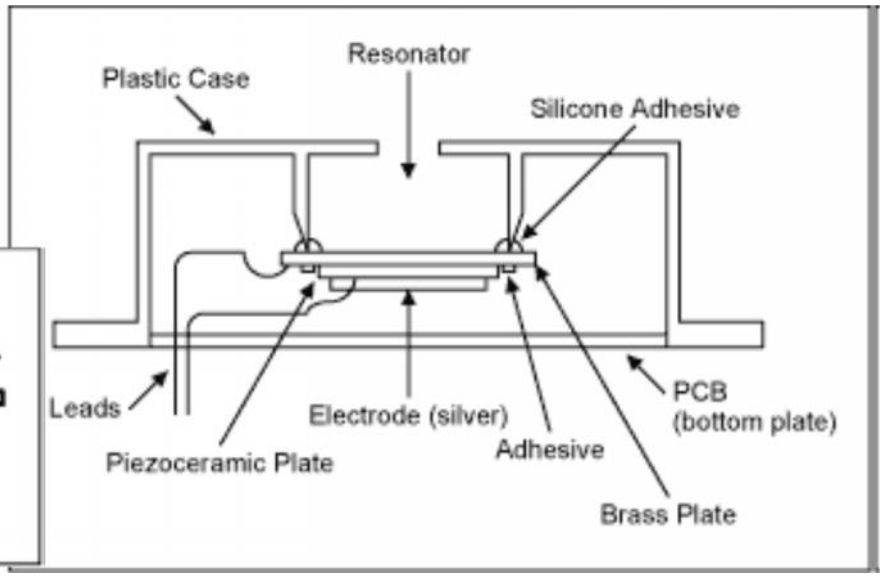
- La parola “**piezein**” in greco significa "spremere" Alcuni cristalli, quando sono sottoposti ad una forte pressione, creano una scintilla.
- Si è scoperto che il processo è reversibile, cioè va anche nel senso opposto quando viene sottoposto ad una tensione ai suoi capi si flette per emettere il suono (flettere qualcosa avanti e indietro permette di muovere l’aria circostante emettendo un suono)

Buzzer piezoelettrici e altoparlanti

- Collegamento tramite 2 conduttori.
- Basta applicare una tensione oscillante per ottenere un suono
- Il buzzer supporta l'elemento piezoelettrico ed ha una cavità risonante per il suono.
- Occorre applicare una tensione fluttuante perché se vien utilizzata solo un livello costante "high" oppure "low" non funzionerà.



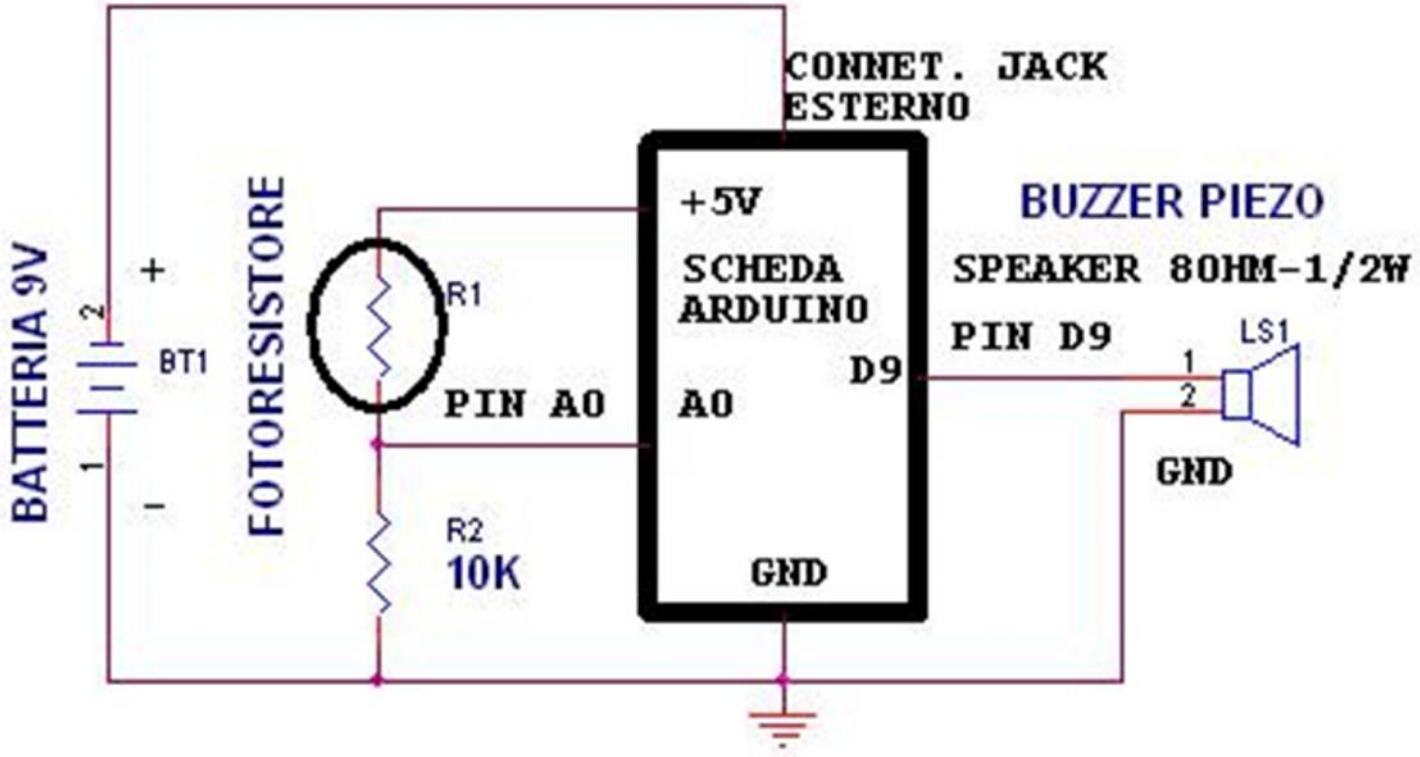




Allarme DIN-DON
per apertura cassetto o frigorifero

Si richiede l'implementazione di un software in grado di:

Generare due note (din-don con uscita digitale ad onda quadra) tramite buzzer o altoparlante quando viene illuminata dalla luce ambiente la fotoresistenza



Allarme DIN-DON per apertura cassetto o frigorifero

```
/*  Allarme DIN-DON per apertura cassetto o frigorifero
collegare un altoparlante da 8 ohm da un terminale sul pin digitale D9
e l'altro terminale a GND. Il fotoresistore R1 tra +5V e l'ingresso
analogico A0, mentre R2 = 10Kohm sara' collegato tra A0 e GND.
La batteria da 9V viene collegata al jack con il polo positivo centrale */

void setup()
{
        pinMode(9, OUTPUT);
}

void loop()
{
        // leggi il valore della tensione sull'input analogico A0
        // finche' tale valore sara' inferiore a circa il 20% di 5V
        // continua a rileggere l'input, altrimenti prosegui
        // perche' la fotoresistenza e' stata colpita dalla luce
        while(analogRead(0) < 200);
        // genera una frequenza di 2000Hz sul pin 9 per 1000ms
        tone(9, 2000, 1000);
        delay(1000);  // aspetta 1000ms
        // genera una frequenza di 440Hz sul pin 9 per 2000ms
        tone(9, 440, 2000);
        delay(2000);  // aspetta 2000ms
        // blocca il generatore di frequenza sul pin 9:
        noTone(9); // disabilita l'altoparlante
}
```

allarme_DIN_DON.ino

Misura della temperatura – LM35

Low Cost

Sensore tipo **LM35** Precision Centigrade Temperature Sensors

Range di temperatura: da **–55** a **+150°C** (contenitore plastico tipo TO-92)

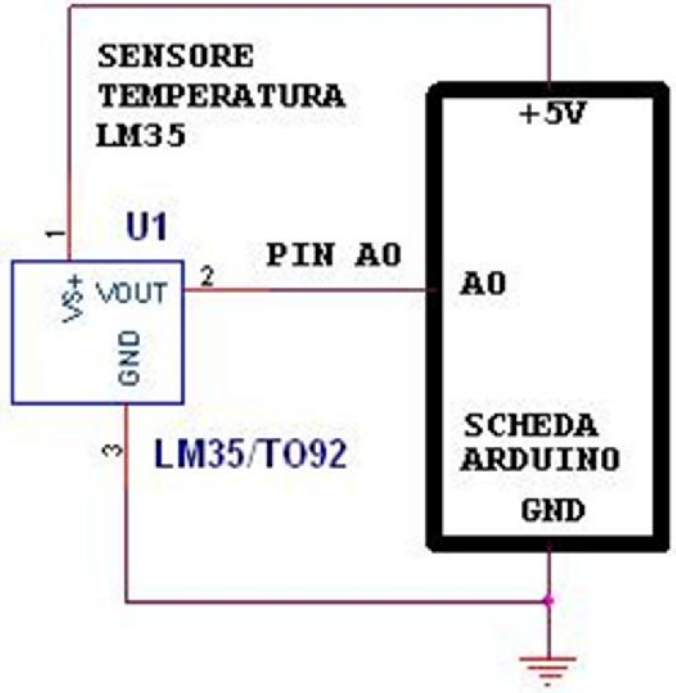
Accuratezza: ±¾°C su tutta la scala, ±1/2°C ambiente)

Bassa impedenza di uscita (0.1 Ω per 1 mA carico)

Uscita in tensione lineare **+10.0 mV/°C**

Alimentazione tra 4 ÷ 30V

Visto di sotto dal lato terminali



Misura della temperatura con LM35

```
/*  I.I.S. Primo LEVI - Torino
  Progetto: Sensore_Temperatura_LM35      Autore: Questo è un esempio di pubblico dominio
  Descrizione: Lettura di un input analogico collegato ad un sensore di temperatura
  del tipo LM35.        Data: 29/01/2012 */
int pin_sensore = 0; // pin 0 di lettura tensione analogica
int temperatura = 0; // variabile temperatura
void setup()
{
 Serial.begin(9600); // inizializza seriale a 9600 baud
}
void loop()
{  // Conversione della tensione generata dal sensore di temperatura
 temperatura = ( 5.0 * analogRead(pin_sensore) * 100.0) / 1024.0;
 Serial.print("Temperatura: "); // trasmetti la scritta "Temperatura: "
 Serial.println(temperatura); // trasmetti il dato della temperatura
 delay(1000); // ritardo tra una lettura e l'altra di 1 sec.
}
```

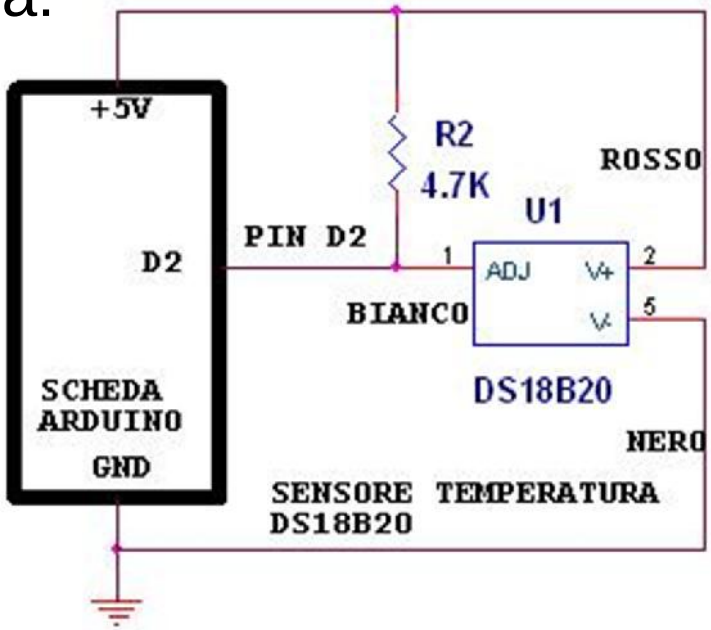
Sensore_Temperatura_LM35.ino

Misura della temperatura – DS18B20

- Sensore tipo **DS18B20** (**1-Wire Digital Thermometer**)
- Range di temperatura: da **–55** a **+125°C** (contenitore waterproof completamente isolato)
- Risoluzione: programmabile da 9 a 12 bit, accuratezza ±1/2°C ambiente
- Uscita digitale: livello TTL compatibile con lettura fino a 32 sensori differenti collegati sulla stessa linea
- Alimentazione tra 3,3 ÷ 5V
- Tempo di conversione della temperatura:

  750 msec. max. a 12 bit

  valori compresi tra 0÷4095.



Misura della temperatura – DS18B20

```
#include <OneWire.h>
int DS18S20_Pin = 2; //DS18S20 Signal pin on digital 2
//Temperature chip i/o
OneWire ds(DS18S20_Pin);  // on digital pin 2
void setup(void) {  Serial.begin(9600);  }
void loop(void)
{
  float temperature = getTemp();
  Serial.println(temperature);
  delay(100); //just here to slow down the output so it is easier to read
}
float getTemp()
{  //returns the temperature from one DS18S20 in DEG Celsius
  byte data[12];
  byte addr[8];
  if ( !ds.search(addr))
  {  //no more sensors on chain, reset search
      ds.reset_search();
      return -1000;
  }
  if ( OneWire::crc8( addr, 7) != addr[7])
  {
      Serial.println("CRC is not valid!");
      return -1000;
  }
```

Sensore_Temperatura_DS18B20.ino

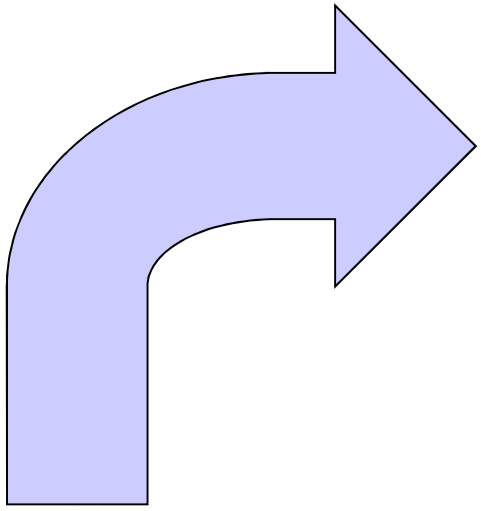
Parte 1ª

Misura della temperatura – DS18B20

```
  if ( addr[0] != 0x10 && addr[0] != 0x28)
  {
      Serial.print("Device is not recognized");
      return -1000;
  }
  ds.reset();
  ds.select(addr);
  ds.write(0x44,1); // start conversion, with parasite power on at the end
  byte present = ds.reset();
  ds.select(addr);
  ds.write(0xBE); // Read Scratchpad
  for (int i = 0; i < 9; i++)
  { // we need 9 bytes
    data[i] = ds.read();
  }
  ds.reset_search();
  byte MSB = data[1];
  byte LSB = data[0];
  float tempRead = ((MSB << 8) | LSB); //using two's compliment
  float TemperatureSum = tempRead / 16;
  return TemperatureSum;
}
```

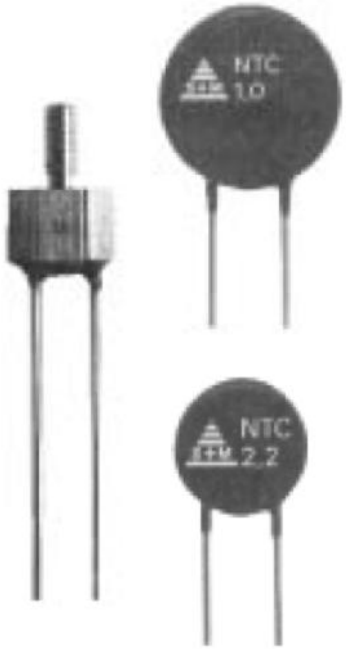
Sensore_Temperatura_DS18B20.ino

Parte 2ª



OUTPUT

```
26.62
26.62
26.62
26.81
26.81
26.81
26.81
26.81
27.00
27.00
27.00
27.00
27.00
27.19
27.19
27.19
27.19
27.19
27.19
27.37
27.37
27.37
27.37
27.37
27.37
27.56
27.56
```



Misura della temperatura con Termistori NTC

- Come definizione un termistore di tipo **NTC** (**Negative Temperature Coefficient**) è un resistore metallico sensibile alla temperatura che fornisce una diminuzione del valore resistivo nominale in conseguenza di un incremento della temperatura.
- Con un valore compreso tra – 2%/K a – 6%/K, si ha un coefficiente negativo della temperatura di circa 10 volte maggiore degli altri metalli.
- L'elevata sensibilità dei termistori NTC è ideale nelle applicazioni di misura della temperatura in cui si richiede un basso costo.
- I sensori NTC vengono utilizzati generalmente entro un range di misura della temperatura compreso tra **–40** e **+300 °C**.

Misura della temperatura con Termistori NTC

NTC.ino - 1ª parte

```
/*  I.I.S. Primo LEVI - Torino
  Progetto: NTC.ino      Autore: Questo è un esempio di pubblico
  Descrizione: Lettura di un input analogico collegato al sensore NTC con il pin A0 e
  l'altro a +5V, mentre un resistore da 1 Kohm verra' collegato tra A0 e massa.
  La visualizzazione della temperatura e' in gradi centigradi sulla seriale */
#include <math.h>
#define ThermistorPIN 0 // collegato al sensore NTC da 5 Kohm Pin analogico 0
float val = 0; // Variabile per il calcolo della temperatura in base al proprio NTC
float NTC = 5000;  // Valore in ohm dell'NTC utilizzato. 5000=5k
float Thermistor(int RawADC)
{
  val = 10000 / NTC;
  val = 4.5 * val;
  val = NTC * val;
  long Resistance;
  float Temp;  // variabile per il calcolo della temperatura

  Resistance=((1024 * val / RawADC) - val);
  Temp = log(Resistance); // salva Long(resistance)
  Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp * Temp * Temp));
  Temp = Temp - 273.15;  // converte gradi Kelvin in Celsius
  return Temp;           // restituisci la temperatura calcolata
}
```

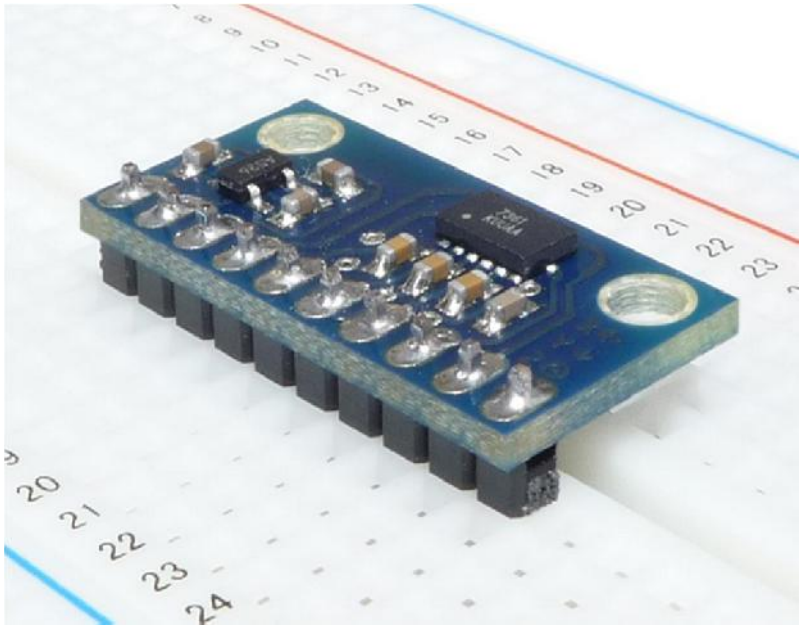
Misura della temperatura con Termistori NTC

NTC.ino - 2ª parte

```
void setup()
{
  Serial.begin(9600); // settaggio velocita' seriale
}

void loop()
{
  float templ;
  templ=Thermistor(analogRead(ThermistorPIN)); // legge valore ADC e converte in °C
  Serial.print("Temperatura: ");
  Serial.print(templ,1);  // Visualizzazione in gradi Celsius
  Serial.write(176);
  Serial.print("C");
  Serial.print(" / ");
  templ = (templ * 9.0) / 5.0 + 32.0; //Converte in Fahrenheit
  Serial.print(templ,1);  // Visualizza in gradi Fahrenheit
  Serial.write(176);
  Serial.println("F");
  delay(1000); // ritardo di 1sec.
}
```

Misura della accelerazione con sensore MMA7361L su 3 assi X, Y e Z



Sensibilità regolabile a **±1.5g** (porre l'input g-Select = LOW = **0V**) oppure a **±6g** (g-Select = HIGH = **3,3V**)

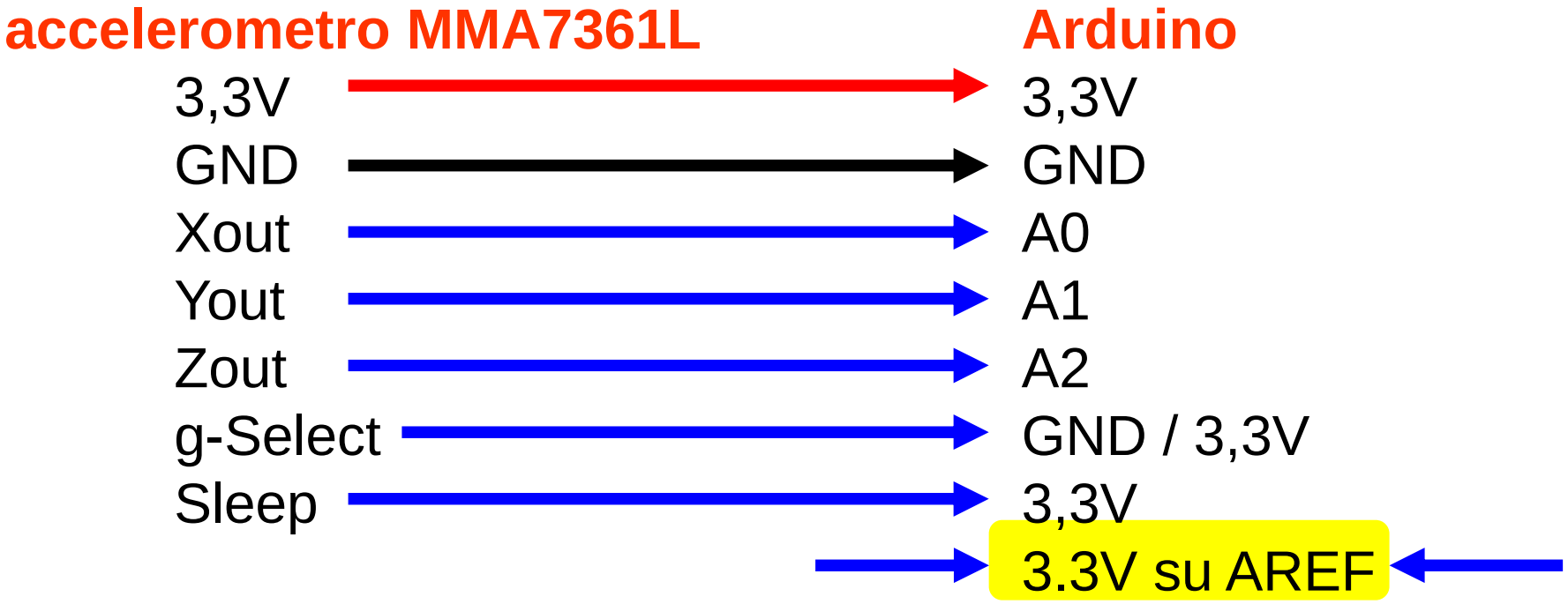
Vdd = +3,3V
Vss = GND
Xout = uscita analogica in mV relativa all'accelerazione dell'asse **X**
Yout = uscita analogica in mV relativa all'accelerazione dell'asse **Y**
Zout = uscita analogica in mV relativa all'accelerazione dell'asse **Z**

Sensore MMA7361L collegamenti

Con la sensibilità impostata a **±1.5g** (porre l'input g-Select = LOW = **0V**) l'accelerazione si può misurarla come variazione dei mV sui singoli pin di out il cui valore varia di **800 mV** ad ogni variazione di **1g** (g = accelerazione di gravità).

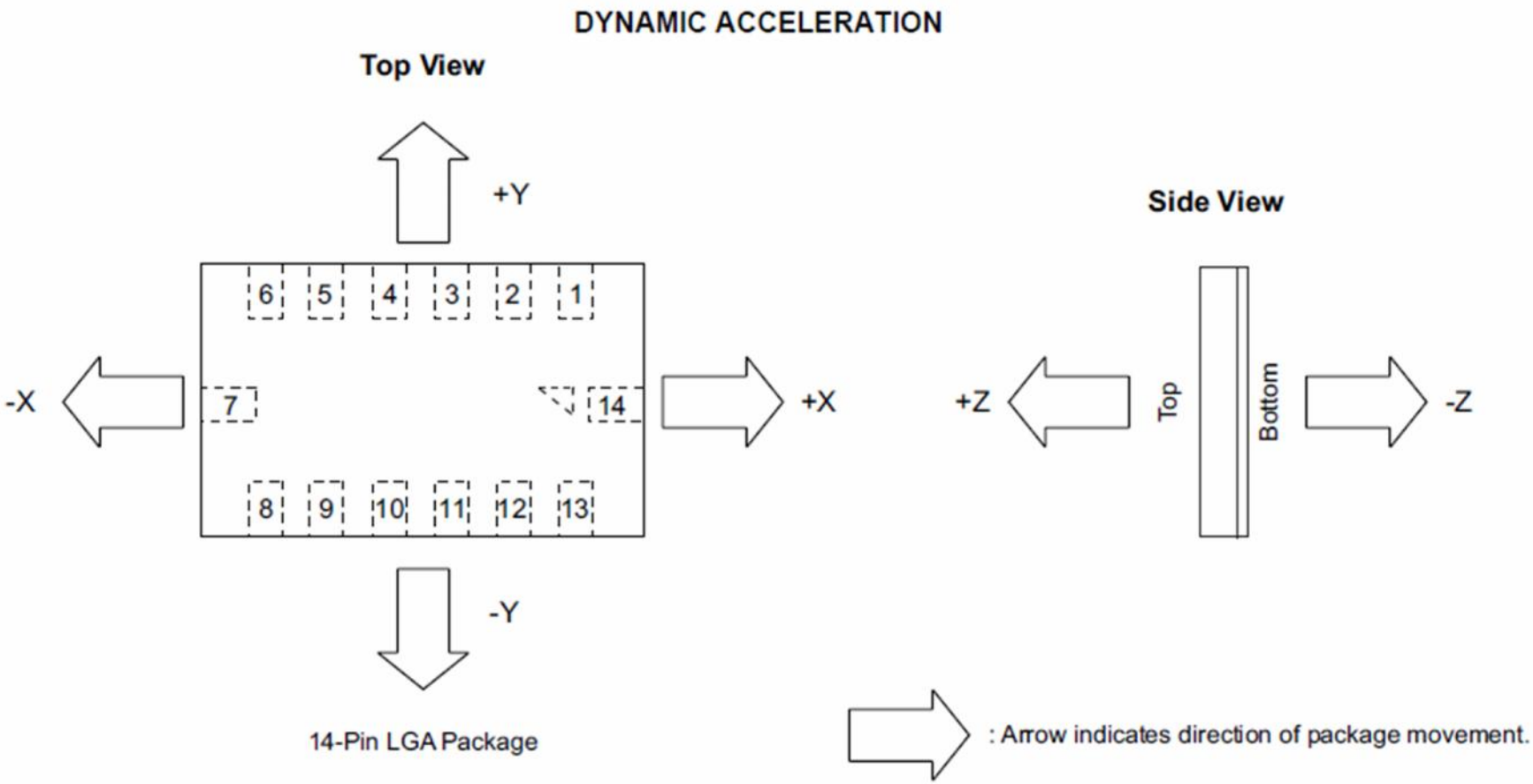
Se la sensibilità è impostata a **±6g** (g-Select = HIGH = **3,3V**) la variazione **ad ogni g** è di soli **206 mV**.

COLLEGAMENTI



Misura della accelerazione con sensore MMA7361L su 3 assi X, Y e Z

Il datasheet dell'accelerometro MMA7361L indica la direzione di ogni asse rispetto all'integrato per aiutarci a comprendere ad ogni movimento gli assi corrispondenti che vengono modificati.



Sensore MMA7361L software

Accelerometro_3_assi_MMA7361L.ino

```
/*  I.I.S. Primo LEVI - Torino    Data: 03/03/2013
  Progetto: accelerometro_3_assi_MMA7361L     Autore: G. Carpignano
  Descrizione: Scrivere un programma in modo tale che venga letto il sensore MMA7361L
  in grado di restituire l'accelerazione sui 3 assi */

int X_pin = A0; // pin analogico dell'Asse X
int Y_pin = A1; // pin analogico dell'Asse Y
int Z_pin = A2; // pin analogico dell'Asse Z

void setup() // inizializzazione, viene eseguito una sola volta all'inizio
{ // inizializza la seriale RS232 con 9600 baud, 8 bit dati, nessuna parità e 1 bit di stop
  Serial.begin(9600);
  analogReference(EXTERNAL); // pin AREF (riferimento analogico esterno) da collegare a 3,3V
  Serial.println("\nInizializzazione del sensore MMA7361L");
}

void loop() // programma principale
{
  Serial.print("X: ");
  Serial.print(analogRead(X_pin) ); // leggi e stampa l'input analogico A0 dell'asse X
  Serial.print("  Y: ");
  Serial.print(analogRead(Y_pin) ); // leggi e stampa l'input analogico A1 dell'asse Y
  Serial.print("  Z: ");
  Serial.print(analogRead(Z_pin) ); // leggi e stampa l'input analogico A2 dell'asse Z
  Serial.print("\n");  // stampa un CR (Carriage Return --> ritorno a inizio riga)
  delay(100);  // ritardo espresso in msec.
}
```

SOFTWARE E SCHEMI ELETTRICI

CON INPUT / OUTPUT

DA COLLEGARE

ALLA SCHEDA ARDUINO

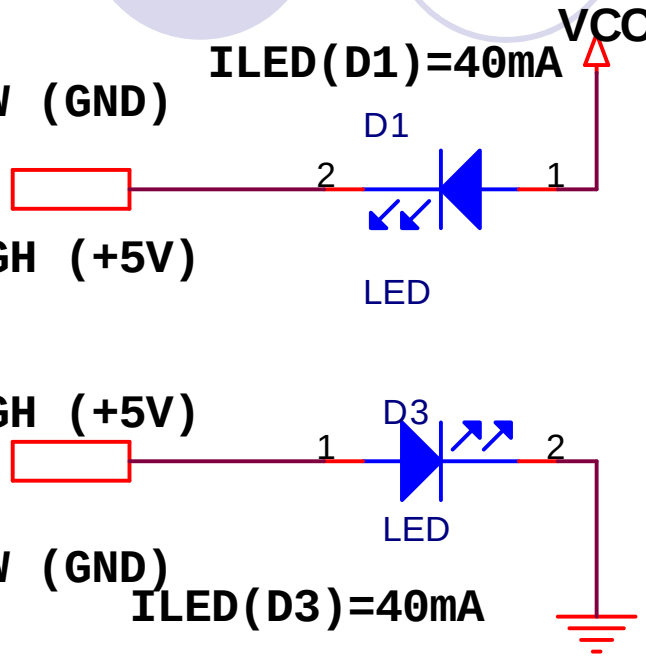
Gestione dei segnalatori ottici (led) con Arduino

LED D1 ACCESO --> PIN13 LOW (GND)

LED D1 SPENTO --> PIN13 HIGH (+5V)

LED D3 ACCESO --> PIN13 HIGH (+5V)

LED D3 SPENTO --> PIN13 LOW (GND)



Versione sconsigliata! (manca la R di limitazione della corrente che scorre nel led, quindi si avrà: ID1 = ID3 = 40mA)

Schema elettrico con 1 led collegato al pin 13 (versione a katodo a massa e anodo al pin 13 senza resistore)

Schema elettrico con 1 led collegato al pin 13 (versione a katodo al pin 13 e anodo a Vcc senza resistore)

Gestione dei segnalatori ottici (led) con Arduino

R1=(VCC-VD2)/ILEDD2=(5-1,8)/0,01=320ohm=330ohm

dove VCC=5V; VD2=1,8V; ILED(D2)=10mA

LED D2 ACCESO --> PIN13 LOW (GND)

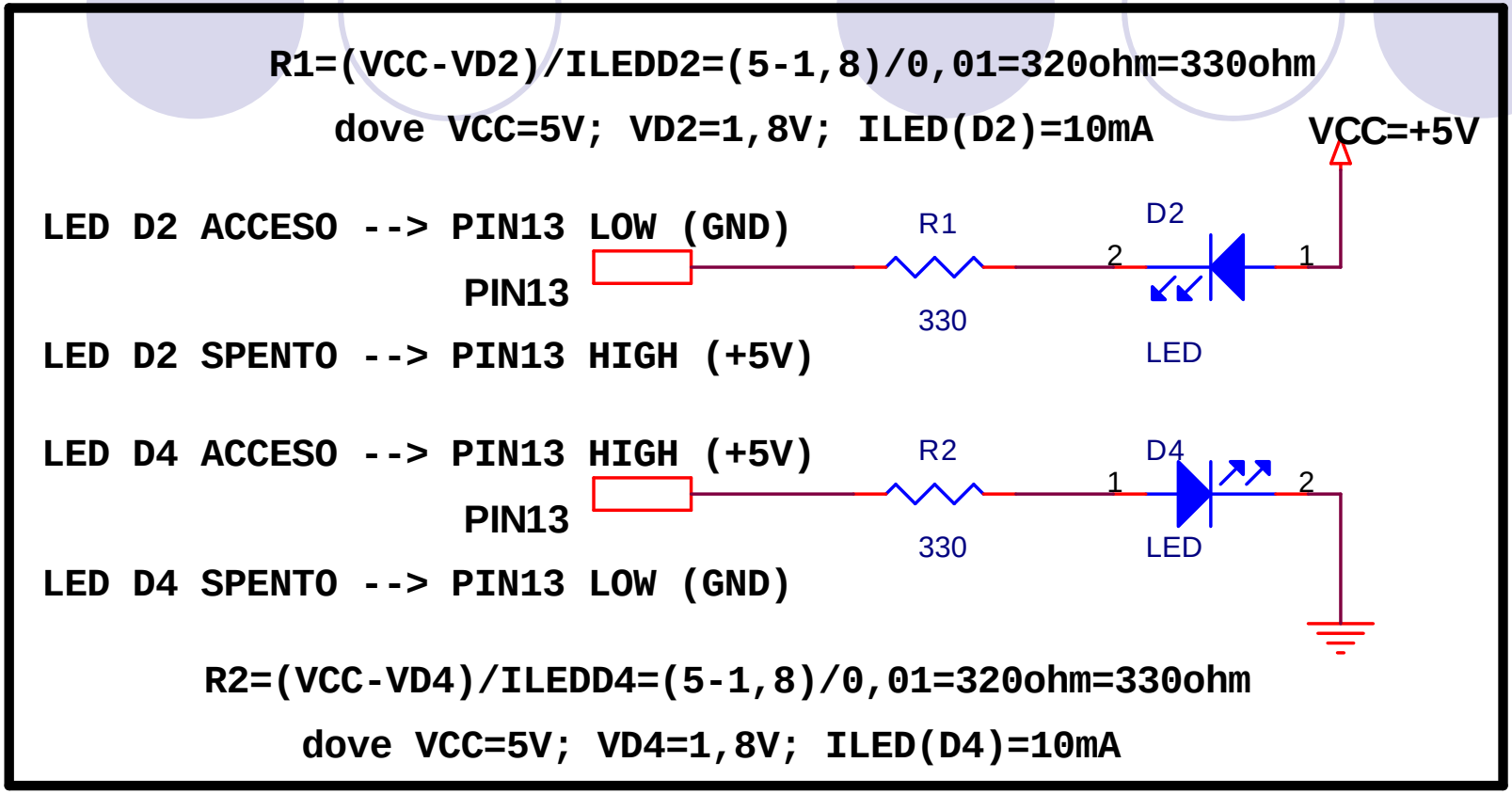
PIN13

LED D2 SPENTO --> PIN13 HIGH (+5V)

LED D4 ACCESO --> PIN13 HIGH (+5V)

PIN13

LED D4 SPENTO --> PIN13 LOW (GND)



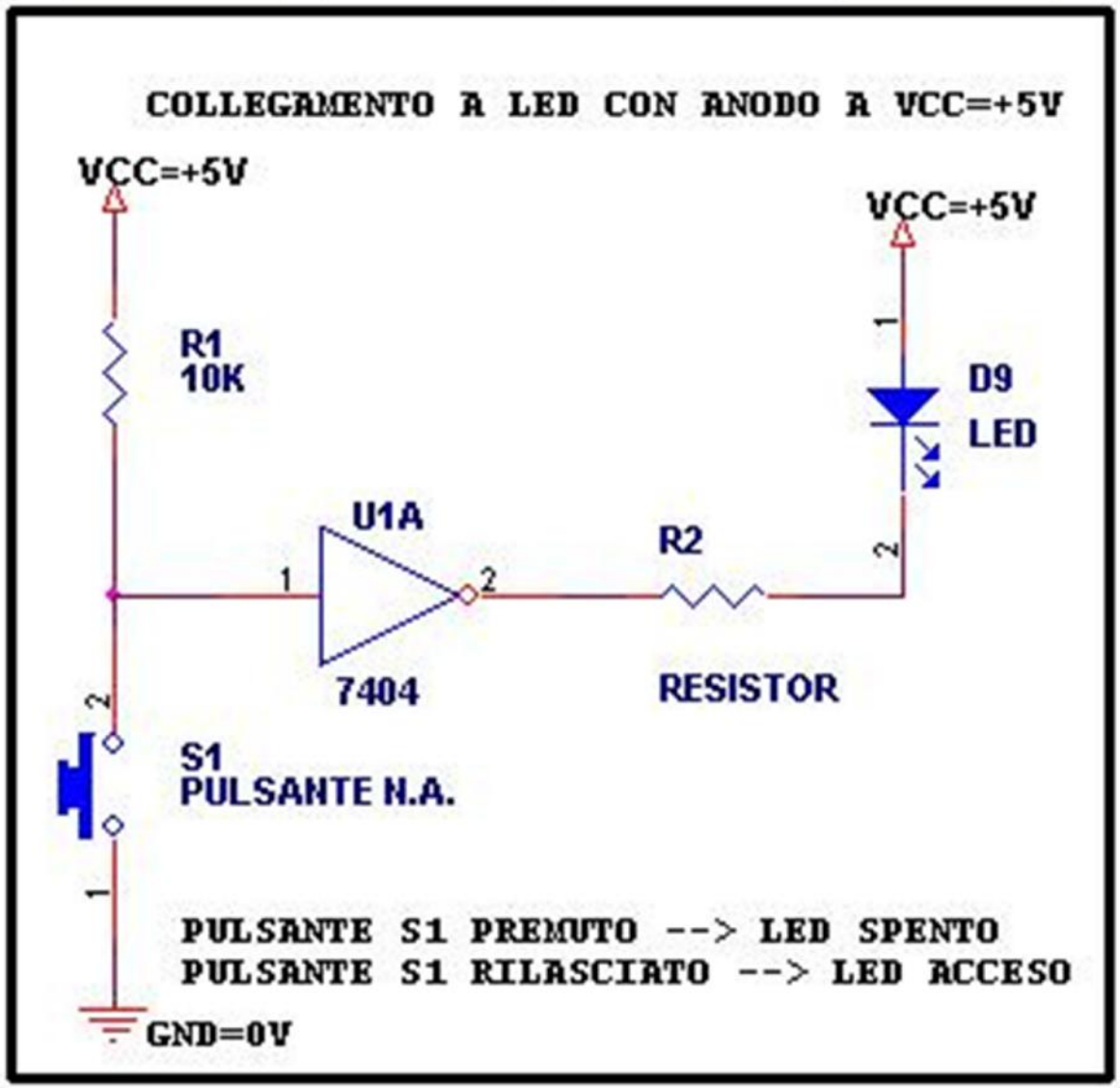
R2=(VCC-VD4)/ILEDD4=(5-1,8)/0,01=320ohm=330ohm

dove VCC=5V; VD4=1,8V; ILED(D4)=10mA

Schema elettrico con 1 led collegato al pin 13 (versione a katodo a massa e anodo al pin 13 con resistore)

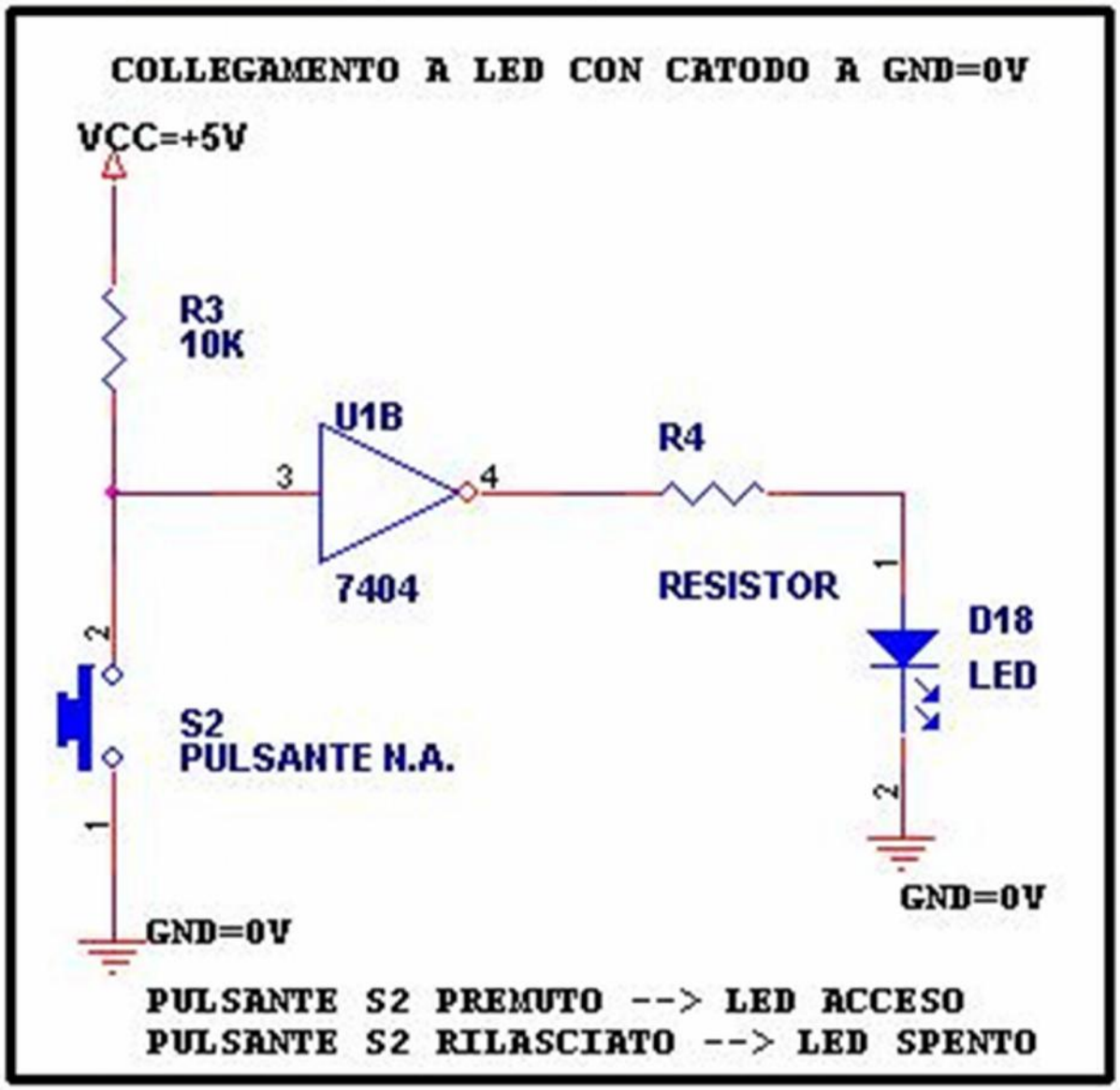
Schema elettrico con 1 led collegato al pin 13 (versione a katodo al pin 13 e anodo a Vcc con resistore)

Gestione dei segnalatori ottici (led) con Arduino



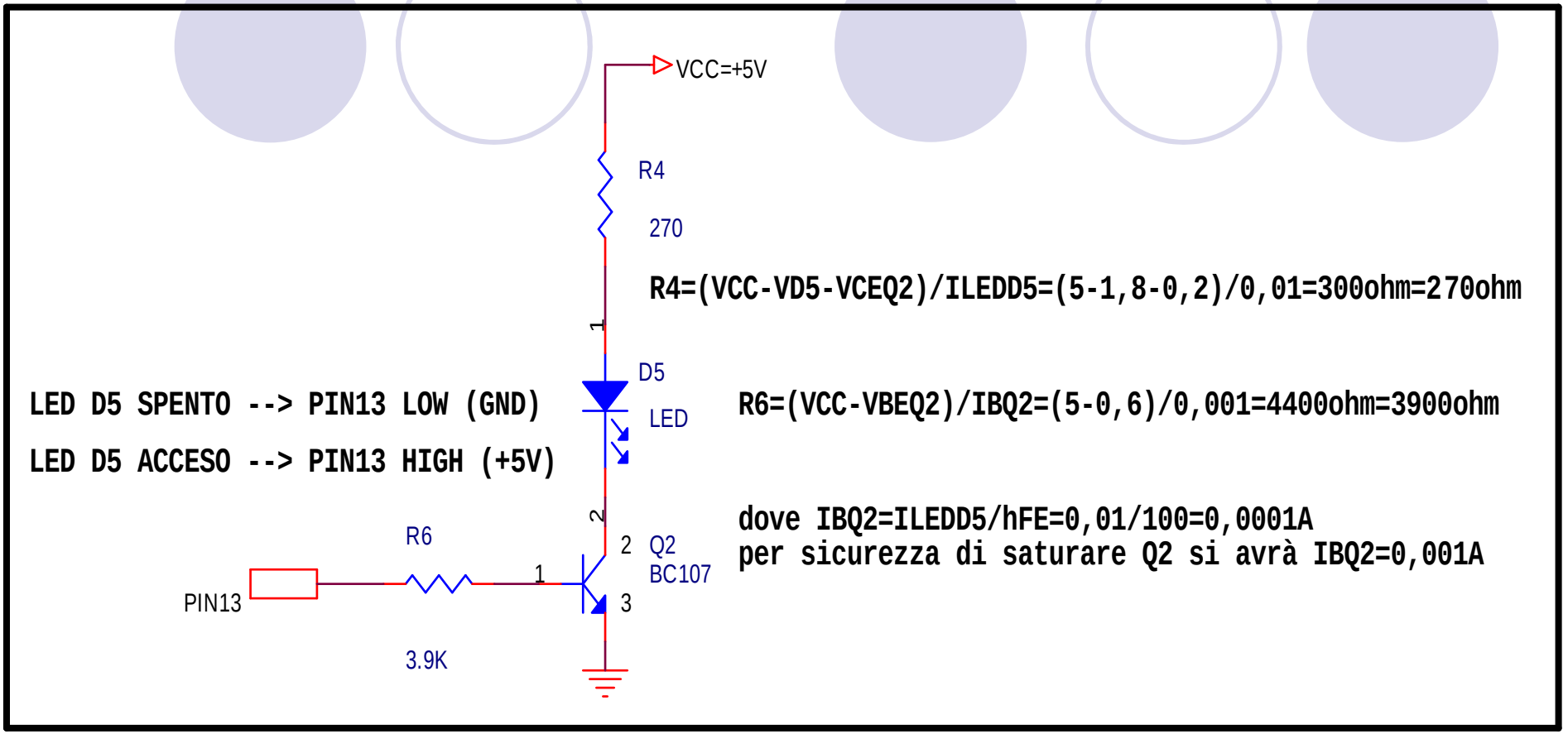
Schema elettrico di un collegamento a led con anodo a VCC=+5V

Gestione dei segnalatori ottici (led) con Arduino



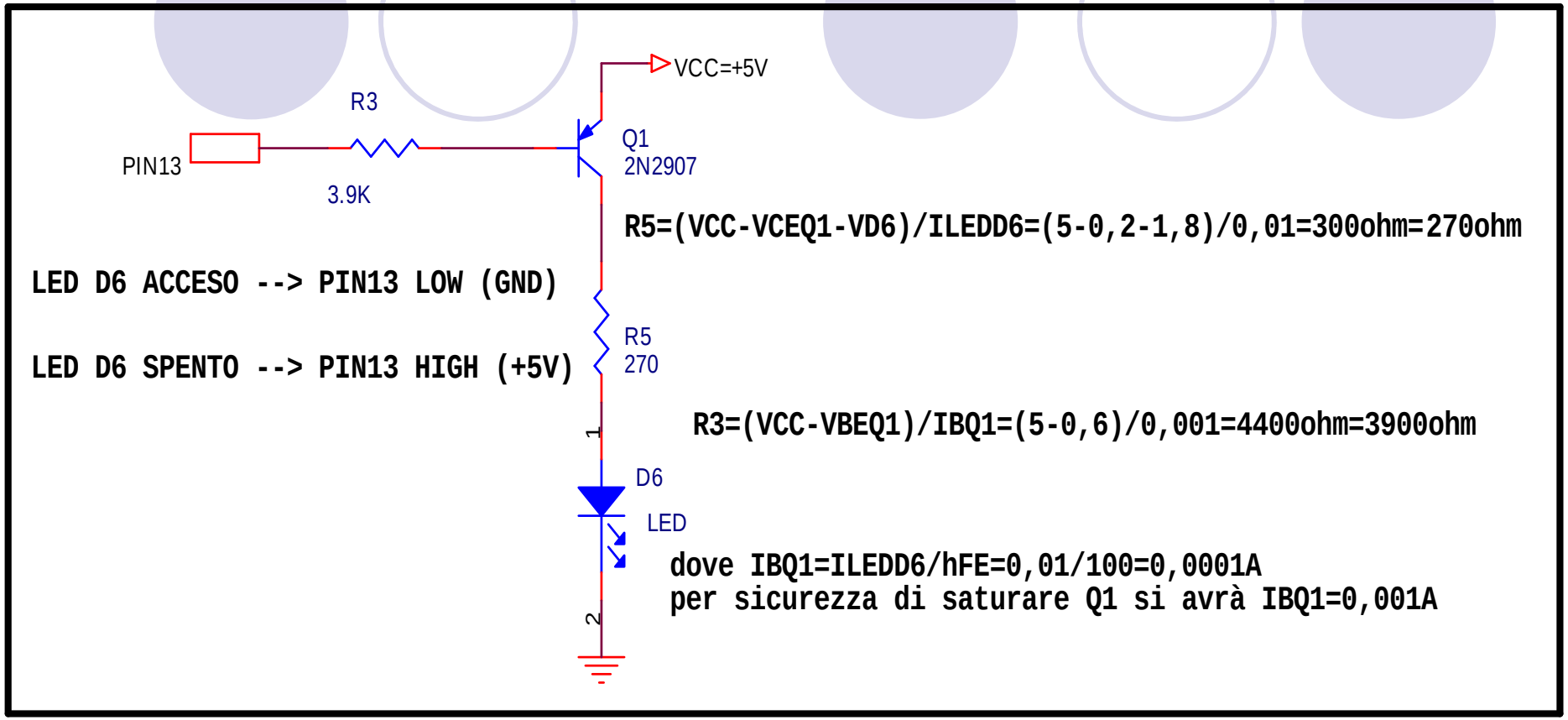
Schema elettrico di un collegamento a led con catodo a GND=0V

Gestione dei segnalatori ottici (led) con Arduino



Schema elettrico con 1 led e un transistor NPN (Vcc=5V)

Gestione dei segnalatori ottici (led) con Arduino



Schema elettrico con 1 led e un transistor PNP (Vcc=5V)

Gestione dei segnalatori ottici (led) con Arduino

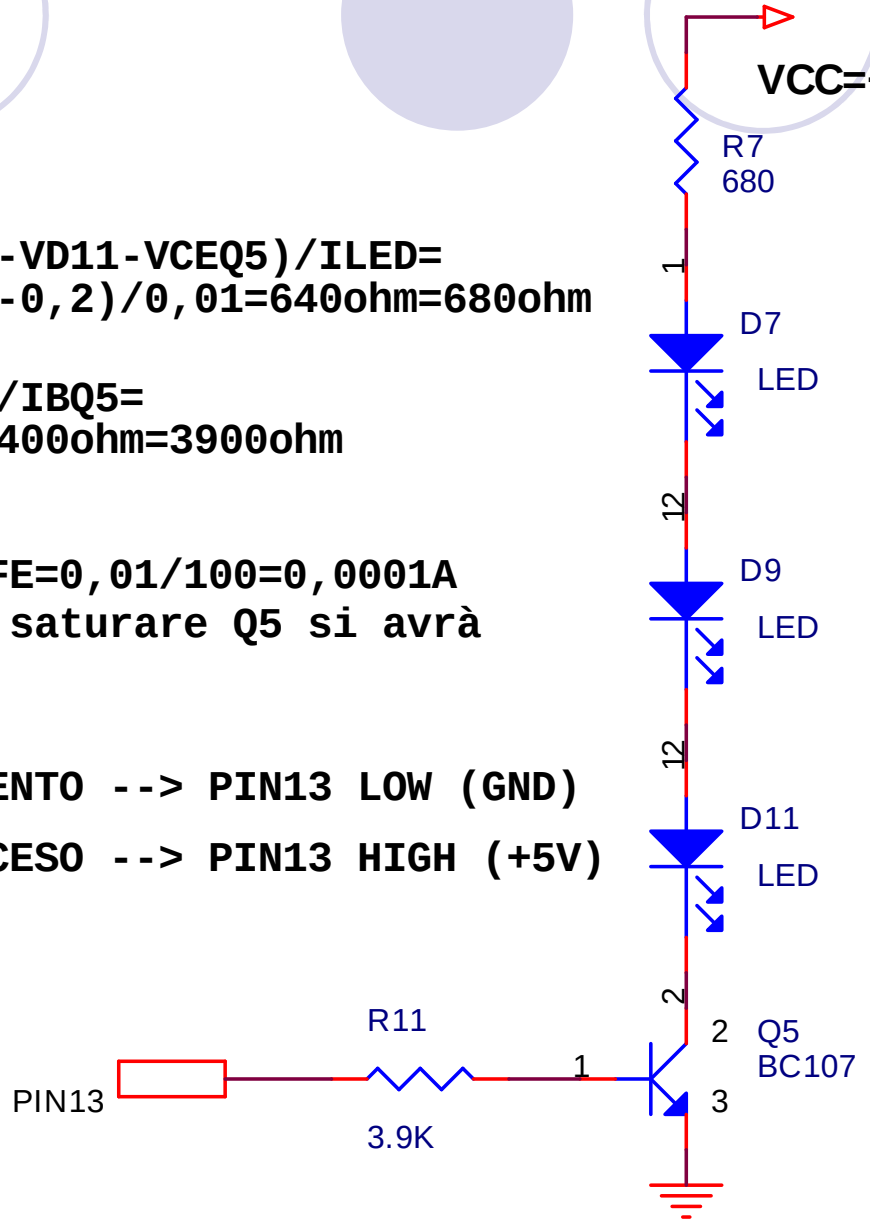
R7=(VCC-VD7-VD9-VD11-VCEQ5)/ILED=
(12-1,8-1,8-1,8-0,2)/0,01=640ohm=680ohm

R11=(VCC-VBEQ5)/IBQ5=
(5-0,6)/0,001=4400ohm=3900ohm

dove IBQ5=ILED/hFE=0,01/100=0,0001A
per sicurezza di saturare Q5 si avrà
IBQ5=0,001A

LED D7,D9,D11 SPENTO --> PIN13 LOW (GND)

LED D7,D9,D11 ACCESO --> PIN13 HIGH (+5V)



Schema elettrico con 3 led e un transistor NPN (Vcc=12V)

Gestione dei segnalatori ottici (led) con Arduino

R9=(VCC-VCEQ3-VD8-VD10-VD12)/ILED=
(12-0,2-1,8-1,8-1,8)/0,01=640ohm=680ohm

R8=(VCC-VBEQ3-VCEQ4)/IBQ3=
(12-0,6-0,2)/0,001=11200ohm=12000ohm

dove IBQ3=ILED/hFE=0,01/100=0,0001A

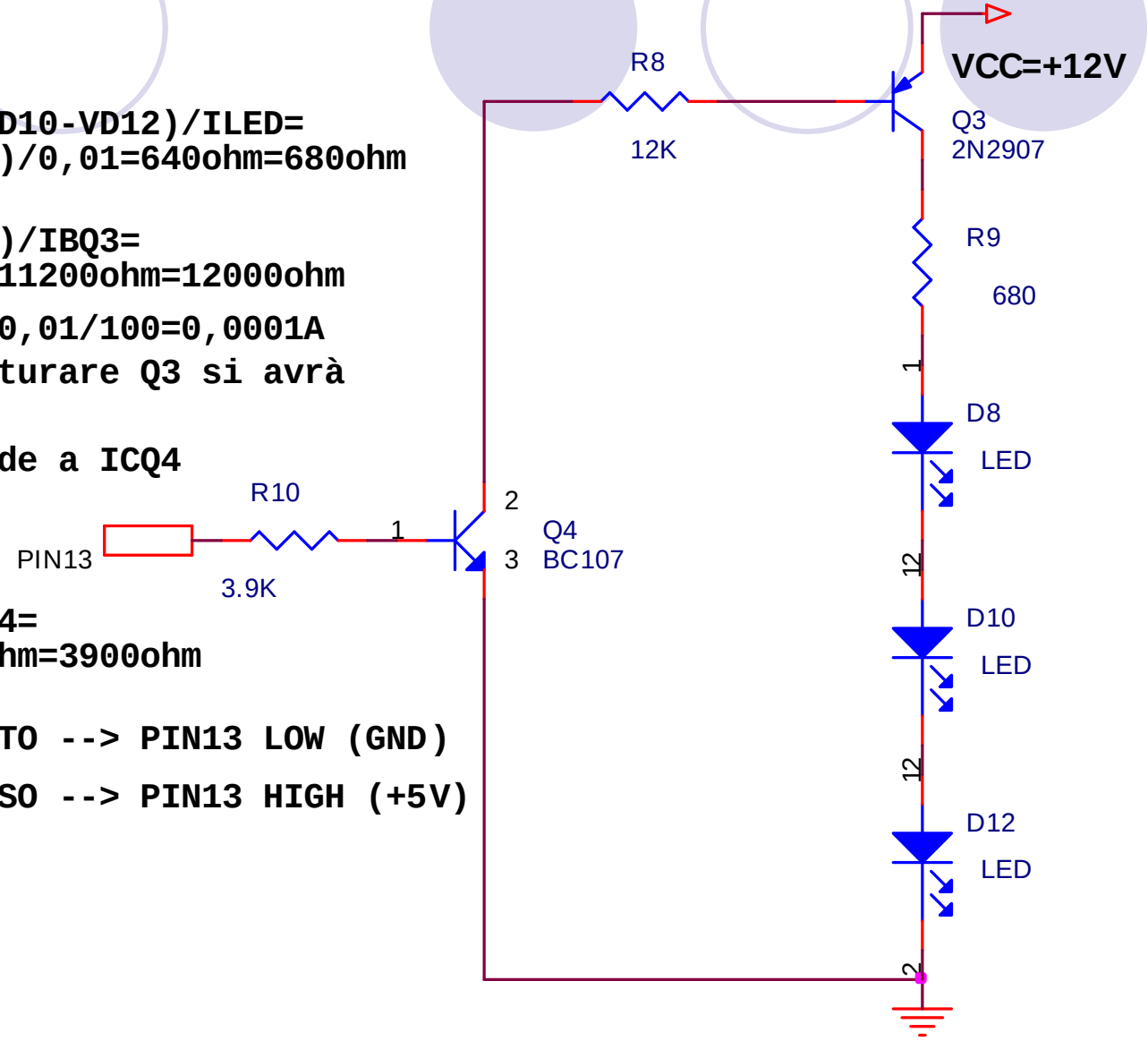
per sicurezza di saturare Q3 si avrà
IBQ3=0,001A

dove IBQ3 corrisponde a ICQ4

R10=(VCC-VBEQ4)/IBQ4=
(5-0,6)/0,001=4400ohm=3900ohm

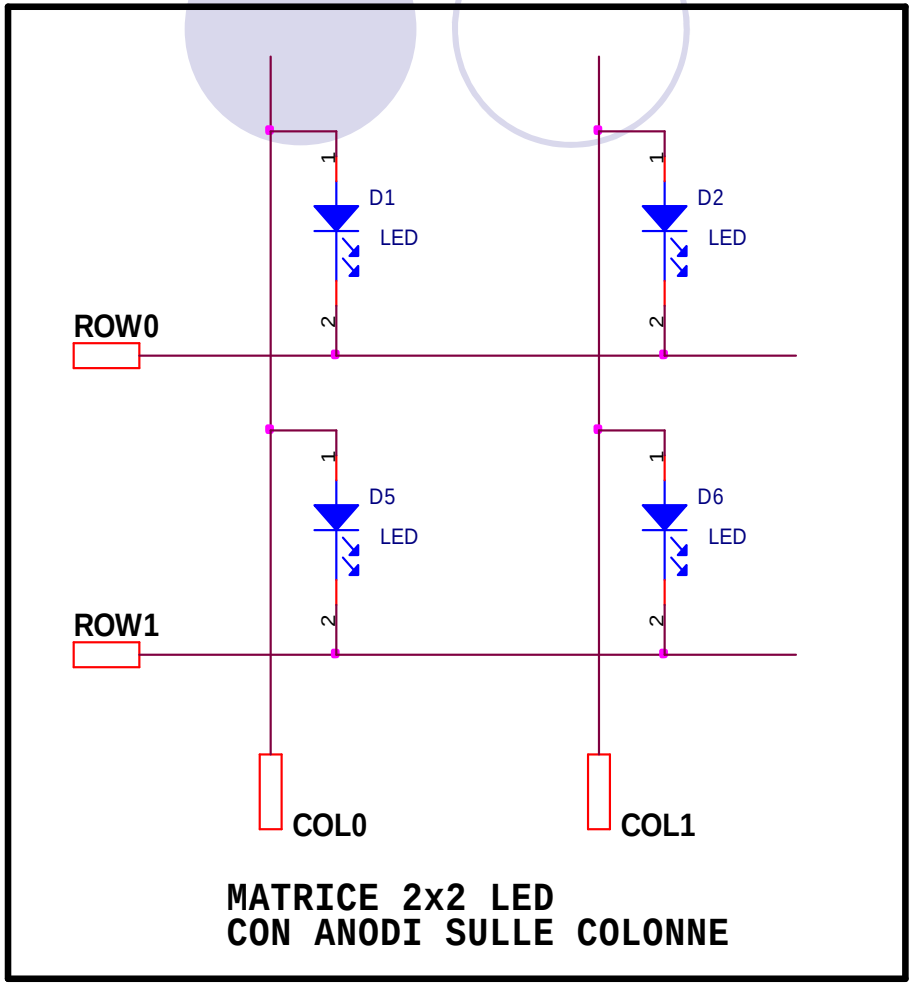
LED D8,D10,D12 SPENTO --> PIN13 LOW (GND)

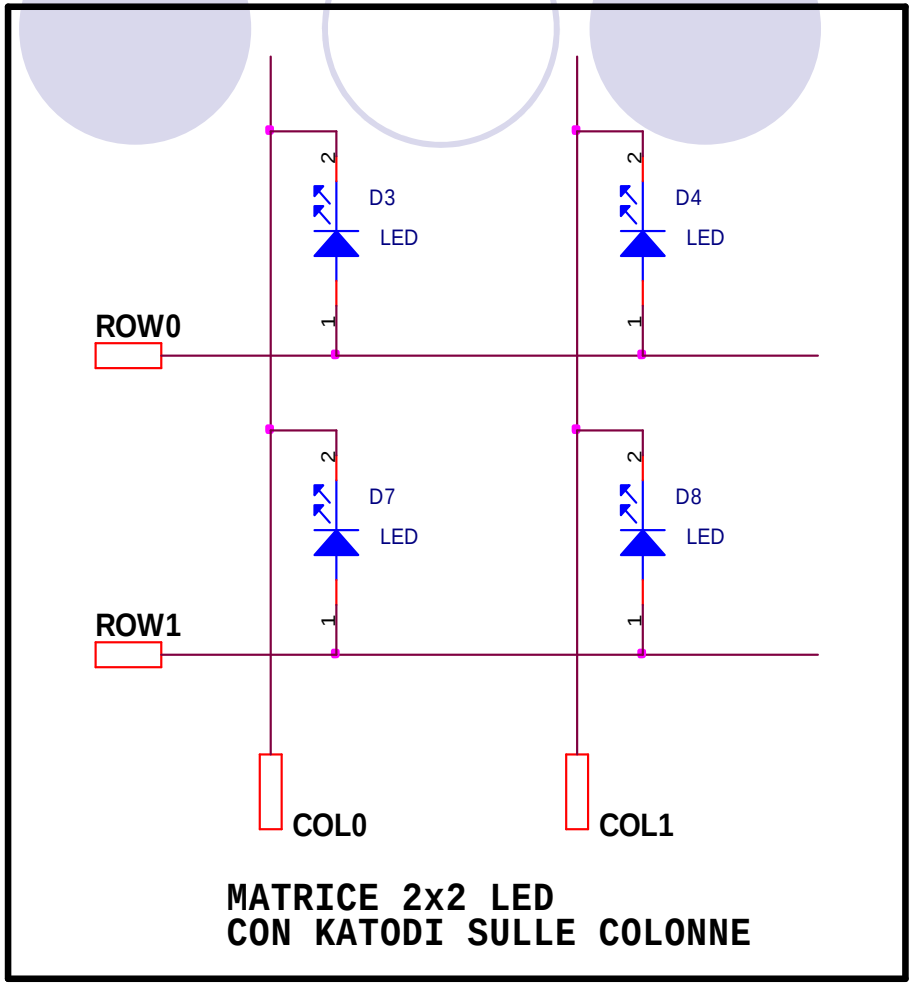
LED D8,D10,D12 ACCESO --> PIN13 HIGH (+5V)



Schema elettrico con 4 led e un transistor NPN+PNP (Vcc=12V)

Gestione dei segnalatori ottici (led) con Arduino



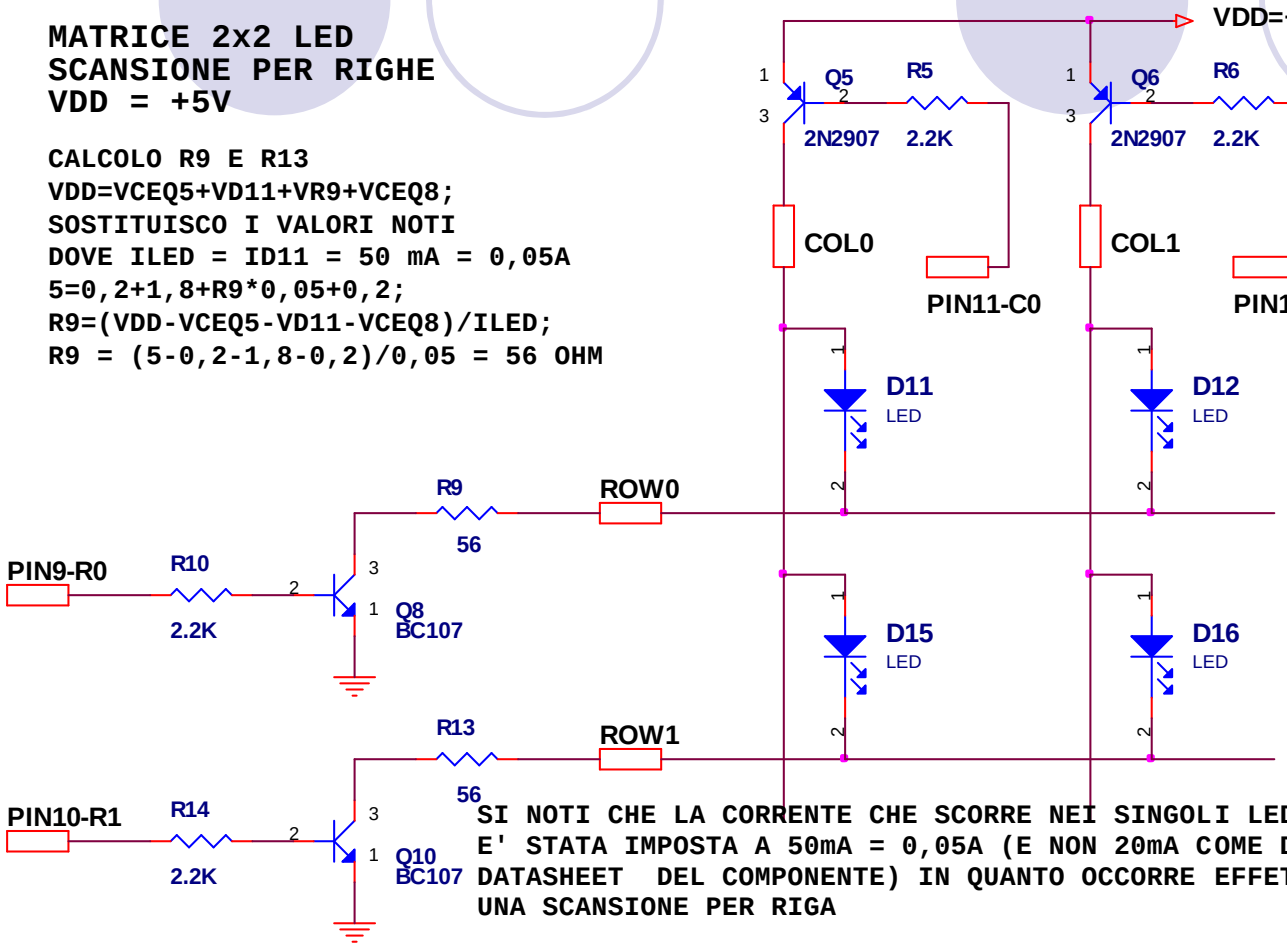


Schema elettrico di una matrice 2 x 2 led
(con anodi sulle colonne a sinistra oppure con katodi sulle colonne a destra)

Gestione dei segnalatori ottici (led) con Arduino

MATRICE 2x2 LED
SCANSIONE PER RIGHE
VDD = +5V

CALCOLO R9 E R13
VDD=VCEQ5+VD11+VR9+VCEQ8;
SOSTITUISCO I VALORI NOTI
DOVE ILED = ID11 = 50 mA = 0,05A
5=0,2+1,8+R9*0,05+0,2;
R9=(VDD-VCEQ5-VD11-VCEQ8)/ILED;
R9 = (5-0,2-1,8-0,2)/0,05 = 56 OHM



SI NOTI CHE LA CORRENTE CHE SCORRE NEI SINGOLI LED E' STATA IMPOSTA A 50mA = 0,05A (E NON 20mA COME DA DATASHEET DEL COMPONENTE) IN QUANTO OCCORRE EFFETTUARE UNA SCANSIONE PER RIGA

CALCOLO R10 E R14
VCC=VBEQ8+VR10
SOSTITUISCO I VALORI NOTI
DOVE IBQ8=ILED/hFE=0,05/50=0,001;
A GARANZIA DI UNA OTTIMA SATURAZIONE DI Q8
RADDOPPIO LA IBQ8=0,001*2=0,002A;
R10=(VCC-VBEQ8)/IBQ8;
R10= (5-0,6)/0,002 = 2200 OHM

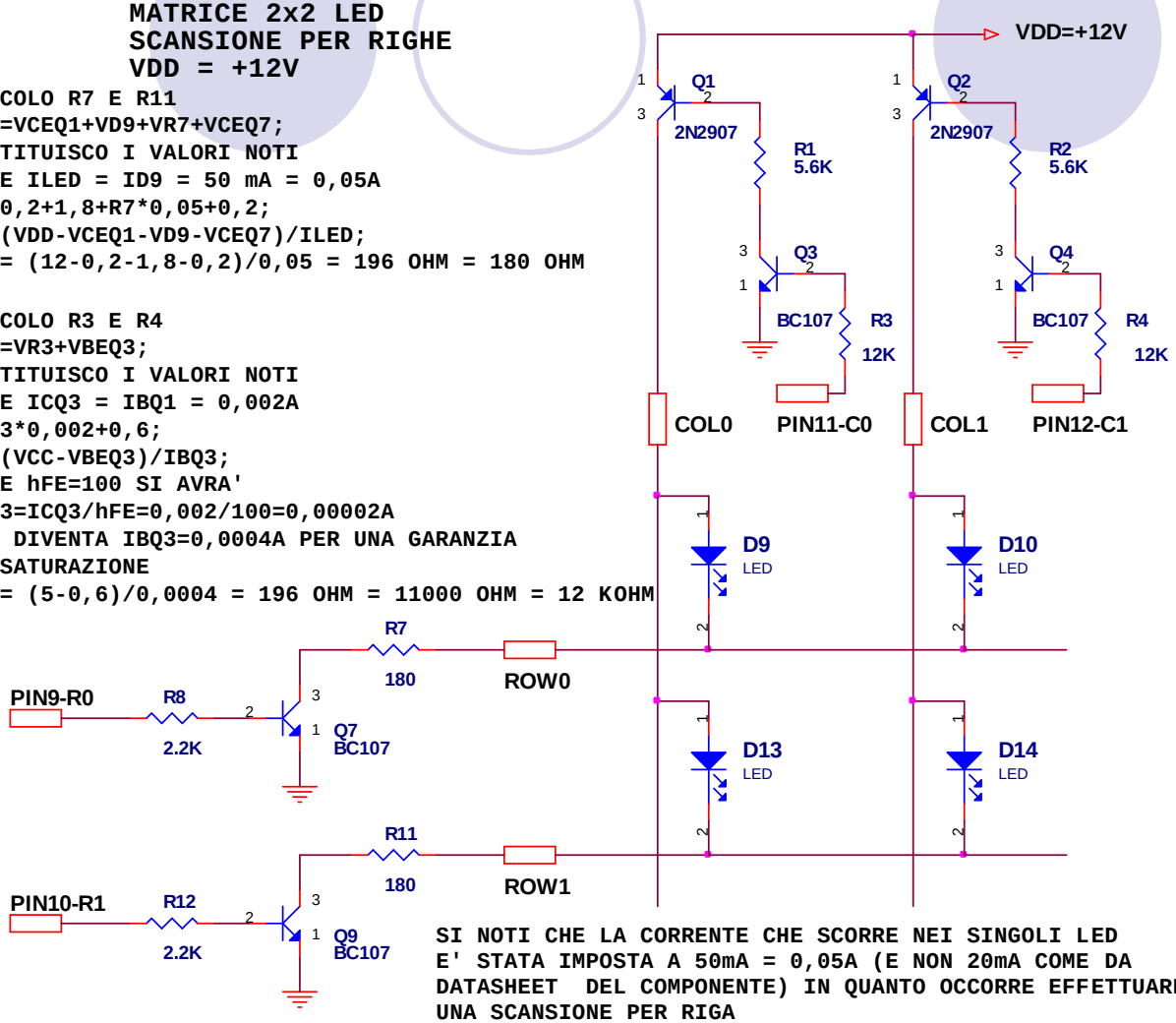
CALCOLO R5 E R6
VDD=VBEQ5+VR5;
SOSTITUISCO I VALORI NOTI
DOVE IBQ5=ILED/hFE=0,05/50=0,001;
A GARANZIA DI UNA OTTIMA SATURAZIONE DI Q5
RADDOPPIO LA IBQ5=0,001*2=0,002A;
R5=(VDD-VBEQ5)/IBQ5;
R5= (5-0,6)/0,002 = 2200 OHM

Schema elettrico di una matrice 2 x 2 led (con anodi sulle colonne alimentata a VDD=+5V) in cui viene effettuata una scansione per righe

Gestione dei segnalatori ottici (led) con Arduino

```
          MATRICE 2x2 LED
          SCANSIONE PER RIGHE
          VDD = +12V
CALCOLO R7 E R11
VDD=VCEQ1+VD9+VR7+VCEQ7;
SOSTITUISCO I VALORI NOTI
DOVE ILED = ID9 = 50 mA = 0,05A
12=0,2+1,8+R7*0,05+0,2;
R7=(VDD-VCEQ1-VD9-VCEQ7)/ILED;
R7 = (12-0,2-1,8-0,2)/0,05 = 196 OHM = 180 OHM

CALCOLO R3 E R4
VCC=VR3+VBEQ3;
SOSTITUISCO I VALORI NOTI
DOVE ICQ3 = IBQ1 = 0,002A
5=R3*0,002+0,6;
R3=(VCC-VBEQ3)/IBQ3;
DOVE hFE=100 SI AVRA'
IBQ3=ICQ3/hFE=0,002/100=0,00002A
CHE DIVENTA IBQ3=0,0004A PER UNA GARANZIA
DI SATURAZIONE
R3 = (5-0,6)/0,0004 = 196 OHM = 11000 OHM = 12 KOHM
```



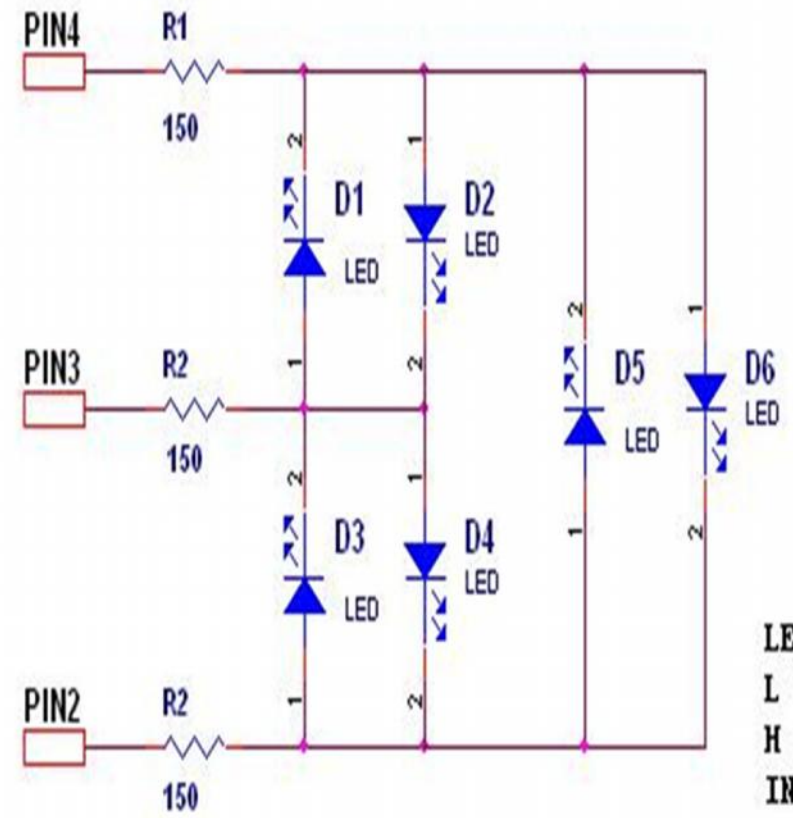

```
SI NOTI CHE LA CORRENTE CHE SCORRE NEI SINGOLI LED
E' STATA IMPOSTA A 50mA = 0,05A (E NON 20mA COME DA
DATASHEET  DEL COMPONENTE) IN QUANTO OCCORRE EFFETTUARE
UNA SCANSIONE PER RIGA

CALCOLO R8 E R12
VCC=VBEQ7+VR8
SOSTITUISCO I VALORI NOTI
DOVE IBQ87=ILED/hFE=0,05/50=0,001;
A GARANZIA DI UNA OTTIMA SATURAZIONE DI Q7
RADDOPPIO LA IBQ7=0,001*2=0,002A;
R8=(VCC-VBEQ7)/IBQ7;
R8= (5-0,6)/0,002 = 2200 OHM

CALCOLO R1 E R2
VDD=VBEQ1+VR5+VCEQ3;
SOSTITUISCO I VALORI NOTI
DOVE IBQ1=ILED/hFE=0,05/50=0,001;
A GARANZIA DI UNA OTTIMA SATURAZIONE DI Q1
RADDOPPIO LA IBQ1=0,001*2=0,002A;
R1=(VDD-VBEQ1-VCEQ3)/IBQ1;
R1= (12-0,6-0,2)/0,002 = 5600 OHM
```

Schema elettrico di una matrice 2 x 2 led (con anodi sulle colonne alimentata a VDD=+12V) in cui viene effettuata una scansione per righe

Gestione dei segnalatori ottici (led) con Arduino



PIN			LEDs						
2	3	4	1	2	3	4	5	6	
L	L	L	0	0	0	0	0	0	--> TUTTI LED SPENTI
L	H	IN	1	0	0	0	0	0	--> LED 1 ACCESO
H	L	IN	0	1	0	0	0	0	--> LED 2 ACCESO
IN	L	H	0	0	1	0	0	0	--> LED 3 ACCESO
IN	H	L	0	0	0	1	0	0	--> LED 4 ACCESO
L	IN	H	0	0	0	0	1	0	--> LED 5 ACCESO
H	IN	L	0	0	0	0	0	1	--> LED 6 ACCESO

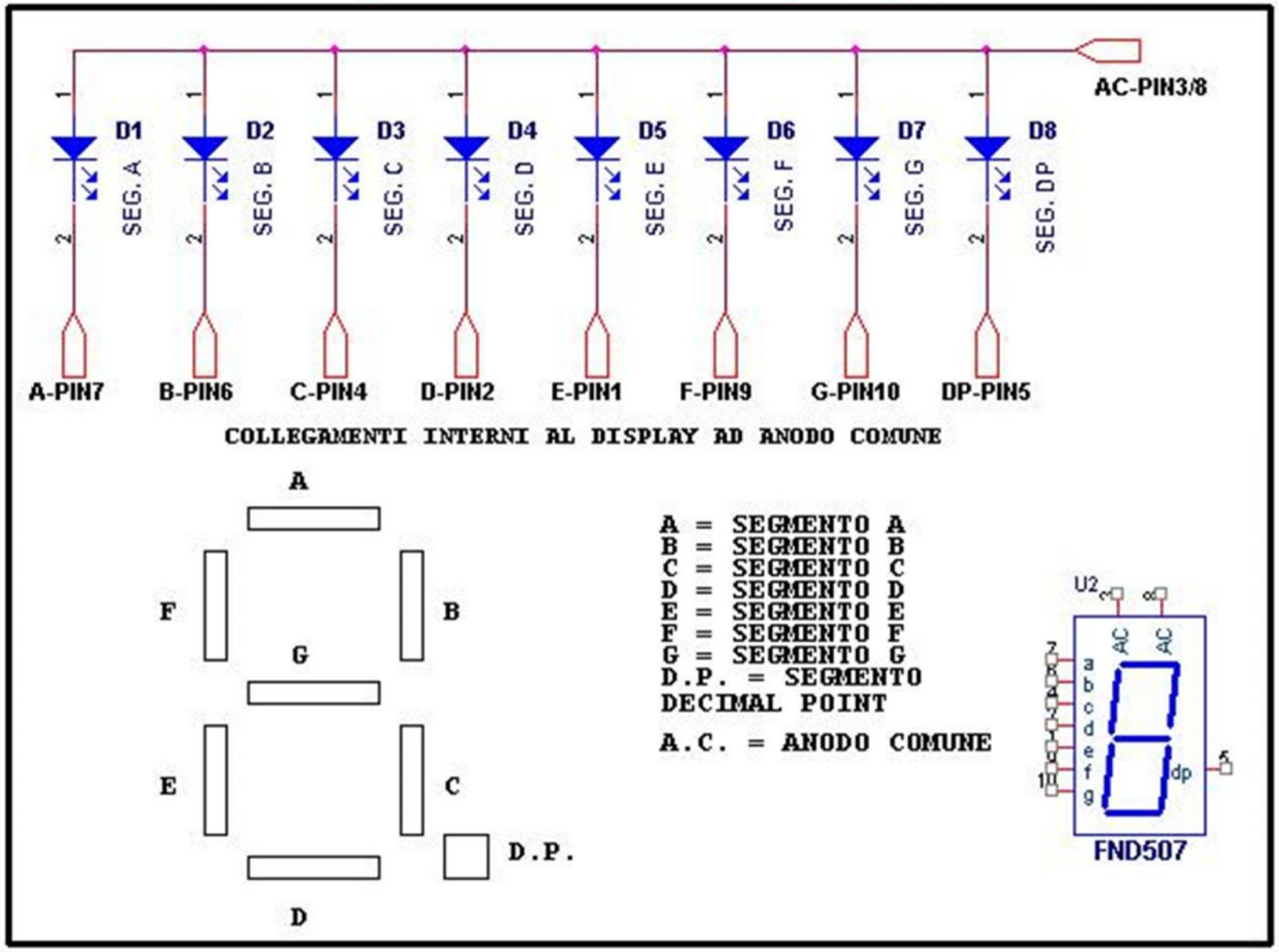
LEGENDA
L = LOW = LIVELLO BASSO = GND = 0V (PIN DEFINITO COME OUTPUT)
H = HIGH = LIVELLO ALTO = VCC = 5V (PIN DEFINITO COME OUTPUT)
IN = PIN DEFINITO COME INPUT

CONTROLLO DI 6 LED CON IL METODO CHARLIEPLEXING

Schema elettrico per il controllo di 6 led tramite 3 pin con il metodo CHARLIEPLEXING.

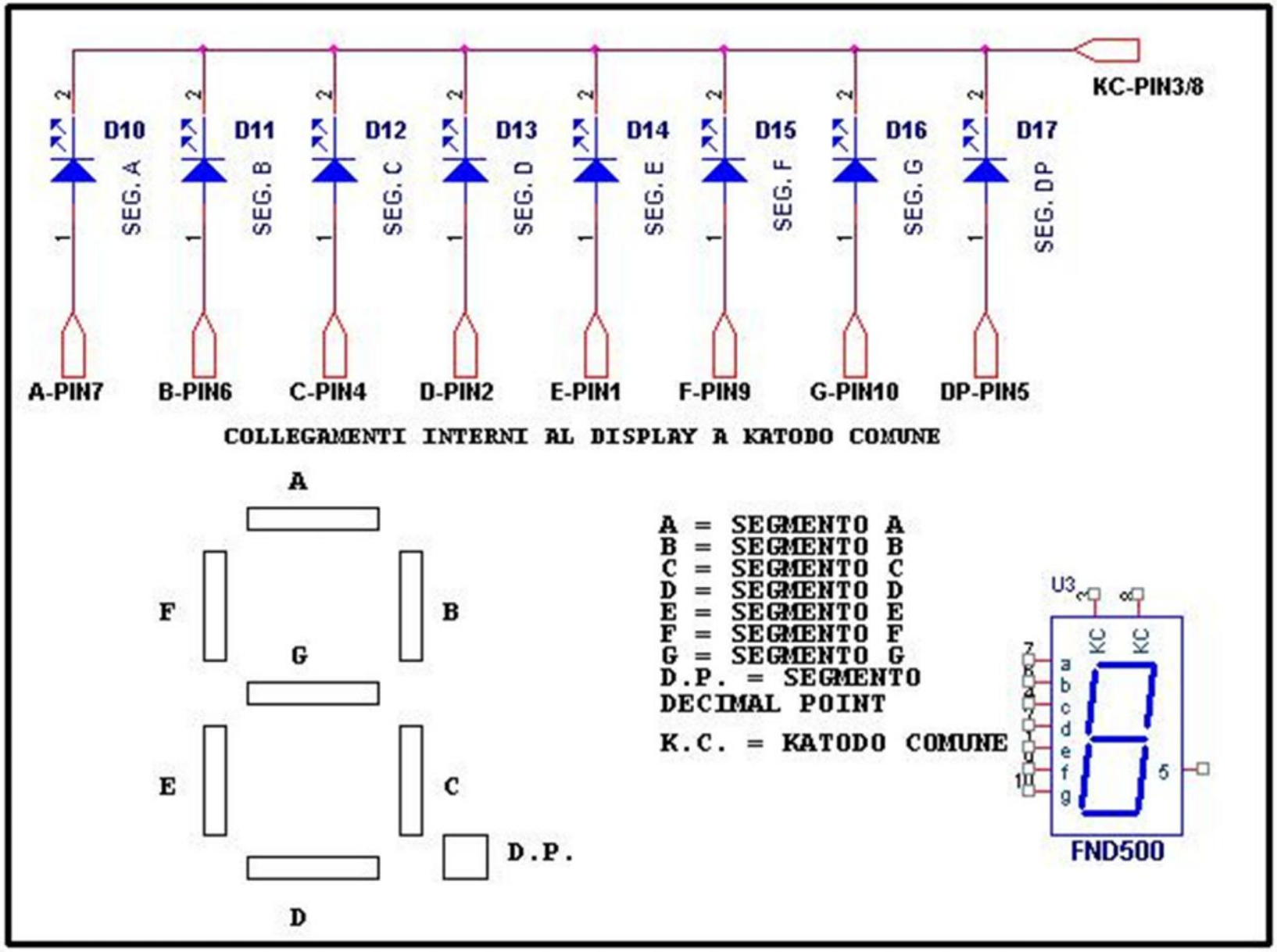
Led_Charlieplexing.ino

Gestione dei segnalatori ottici (led) con Arduino



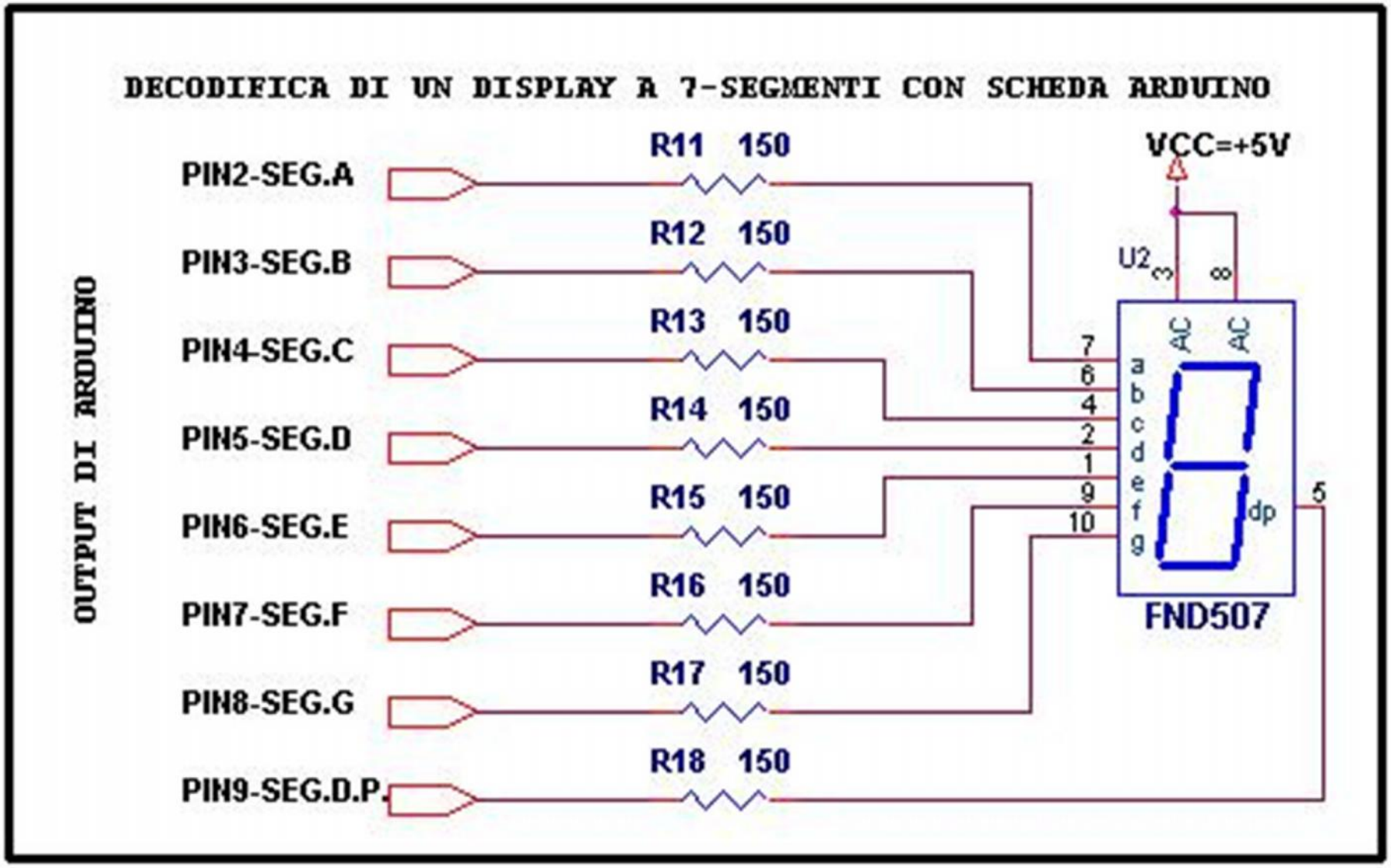
Collegamenti interni ad un display a anodo comune (FND507)

Gestione dei segnalatori ottici (led) con Arduino



Collegamenti interni ad un display a catodo comune (FND500)

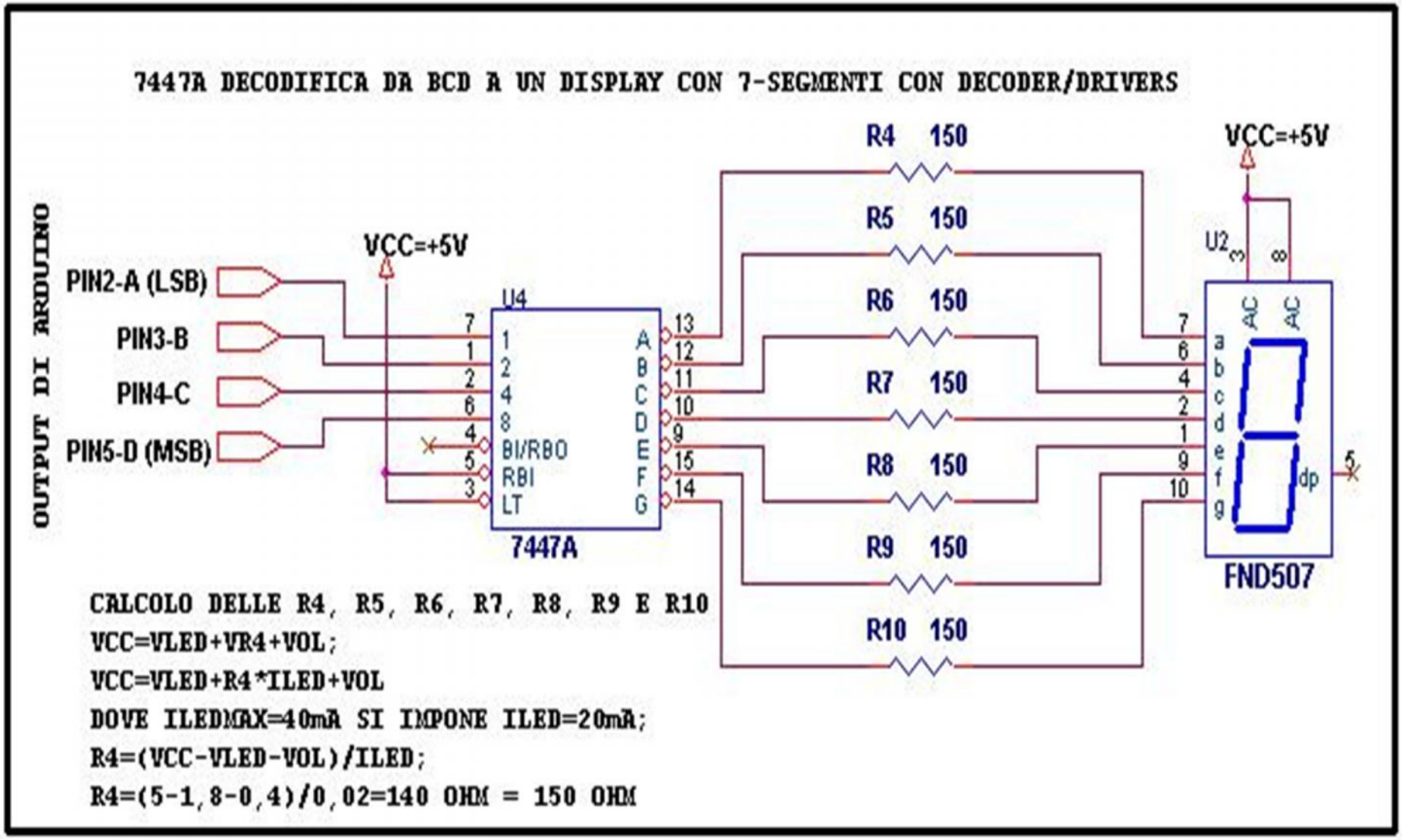
Gestione dei segnalatori ottici (led) con Arduino



display_7_segmenti_anodo_comune.ino

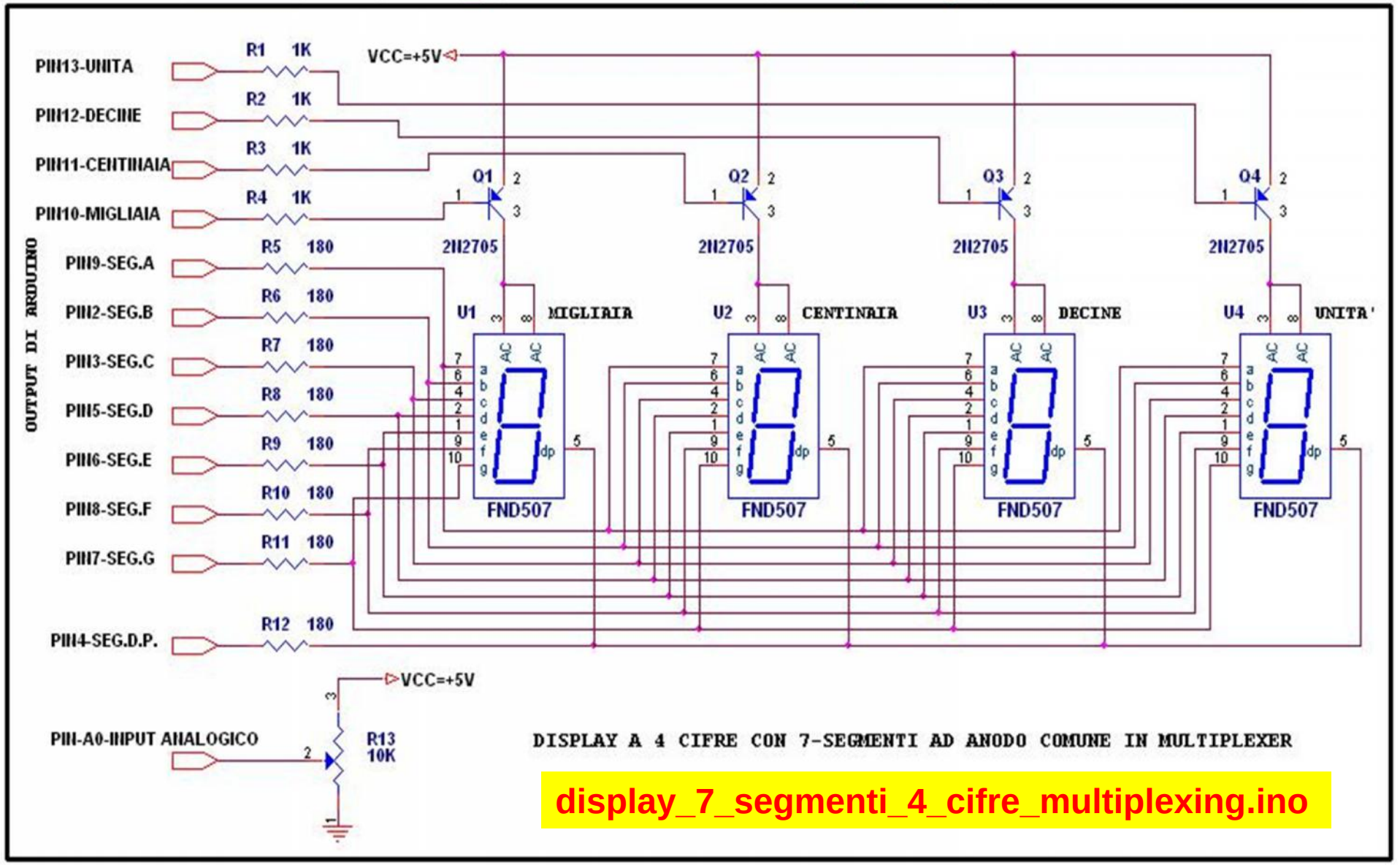
Decodifica di un display a 7 segmenti con la scheda Arduino

Gestione dei segnalatori ottici (led) con Arduino



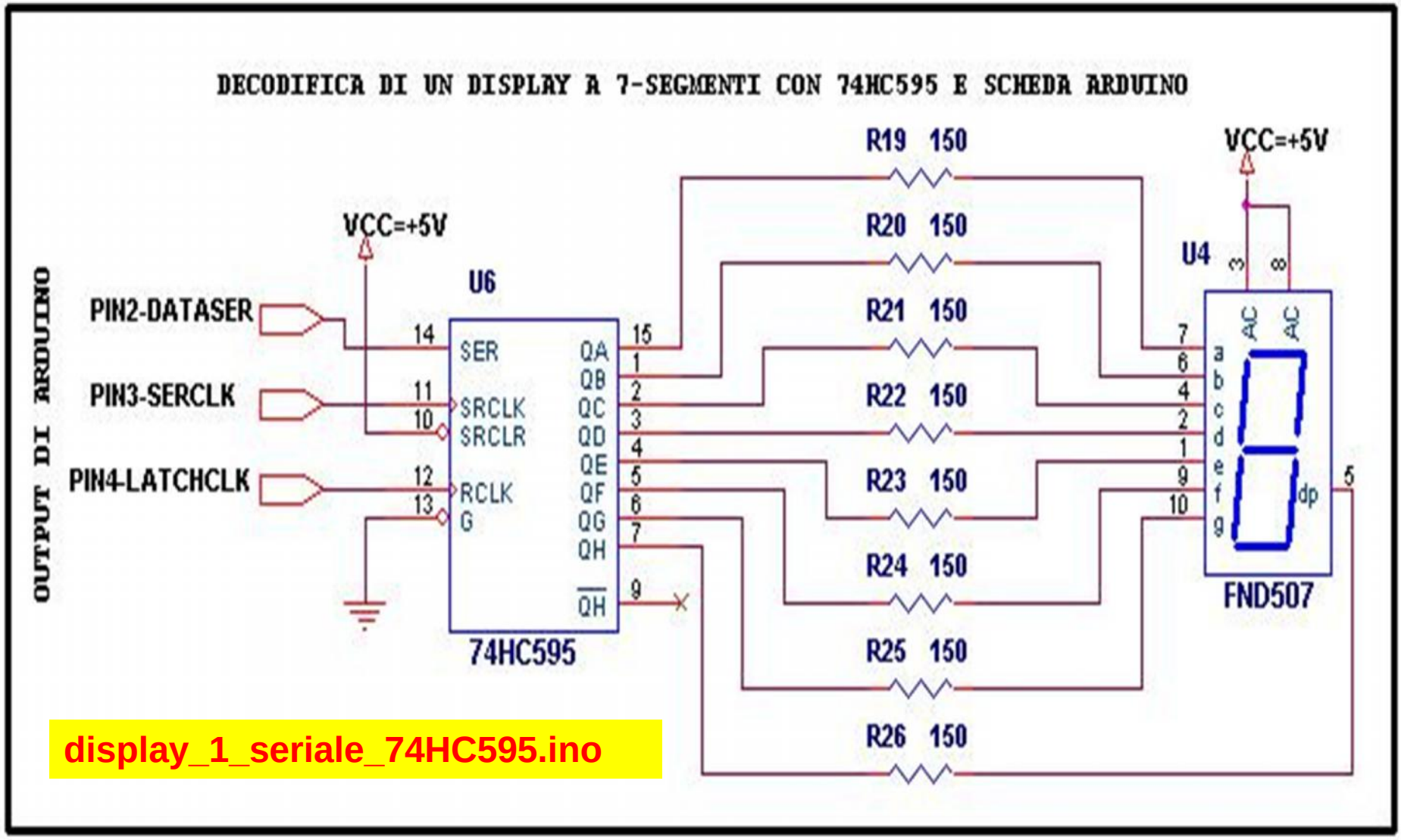
Decodifica da BCD a 7 segmenti con 7447A e scheda Arduino

Gestione dei segnalatori ottici (led) con Arduino



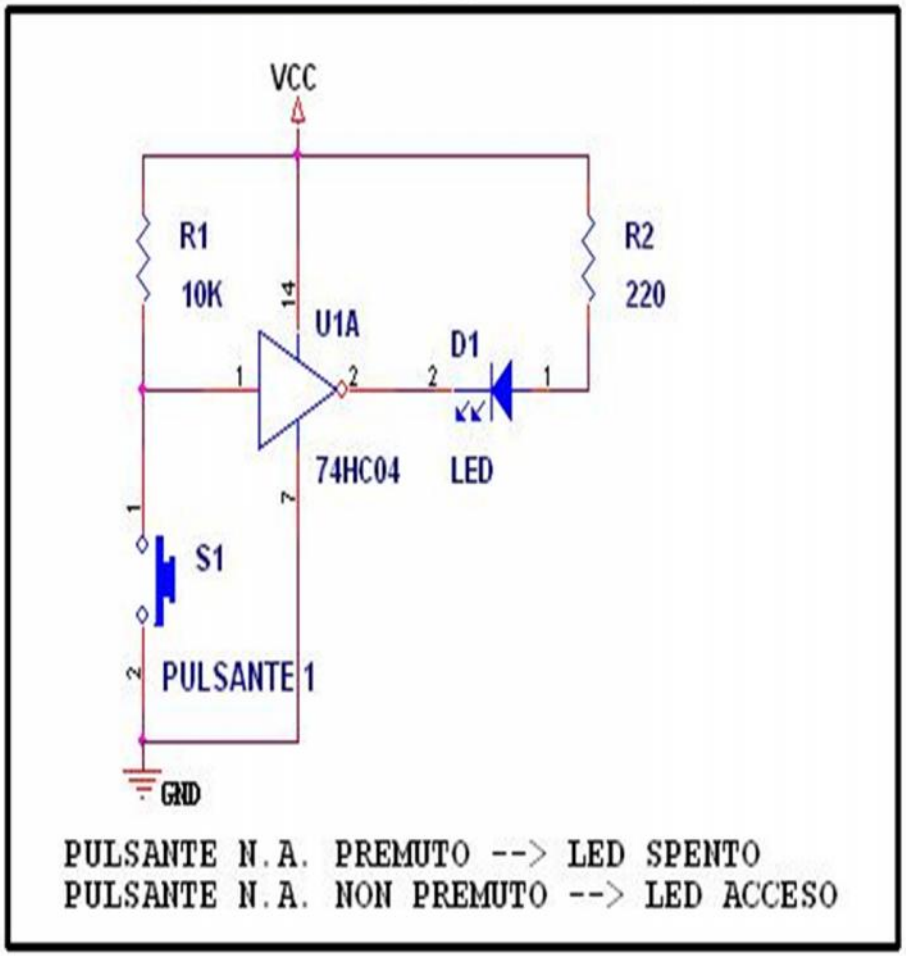
Schema elettrico di un display a 4 cifre con 7 segmenti ad anodo comune da collegare alla scheda Arduino

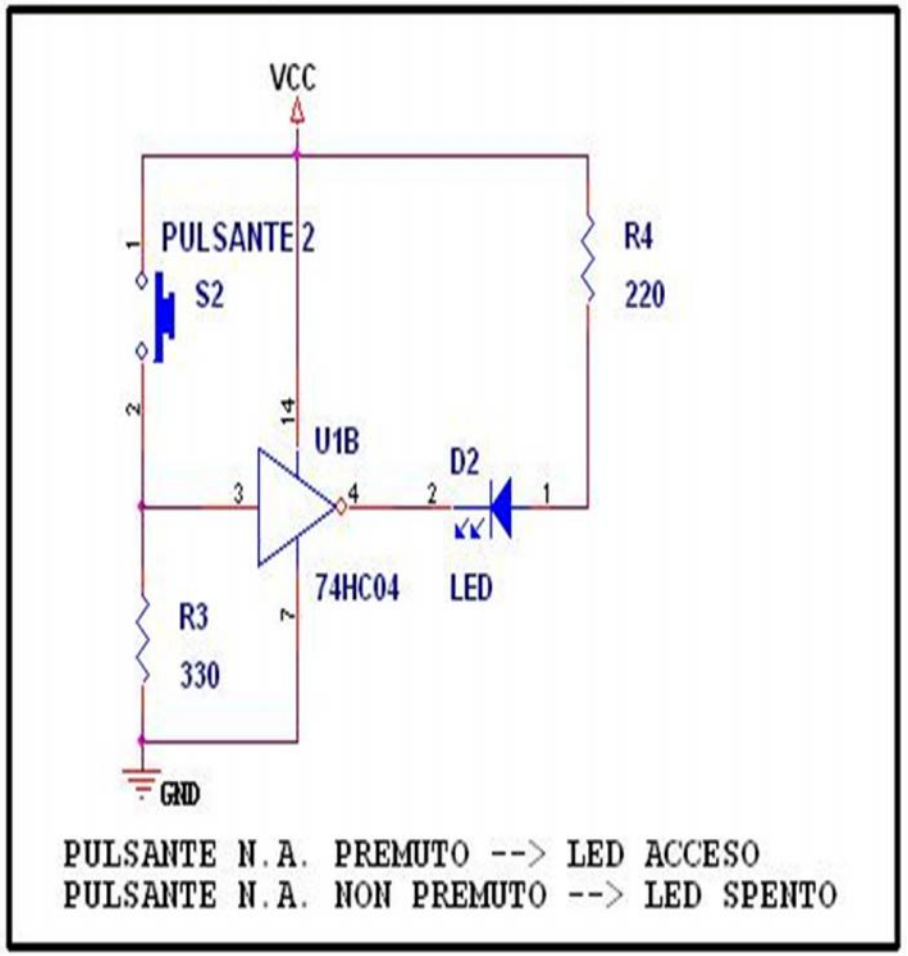
Gestione dei segnalatori ottici (led) con Arduino



Decodifica di un display a 7 segmenti con integrato 74HC595 serial-to-parallel/decoder/latch collegato alla scheda Arduino

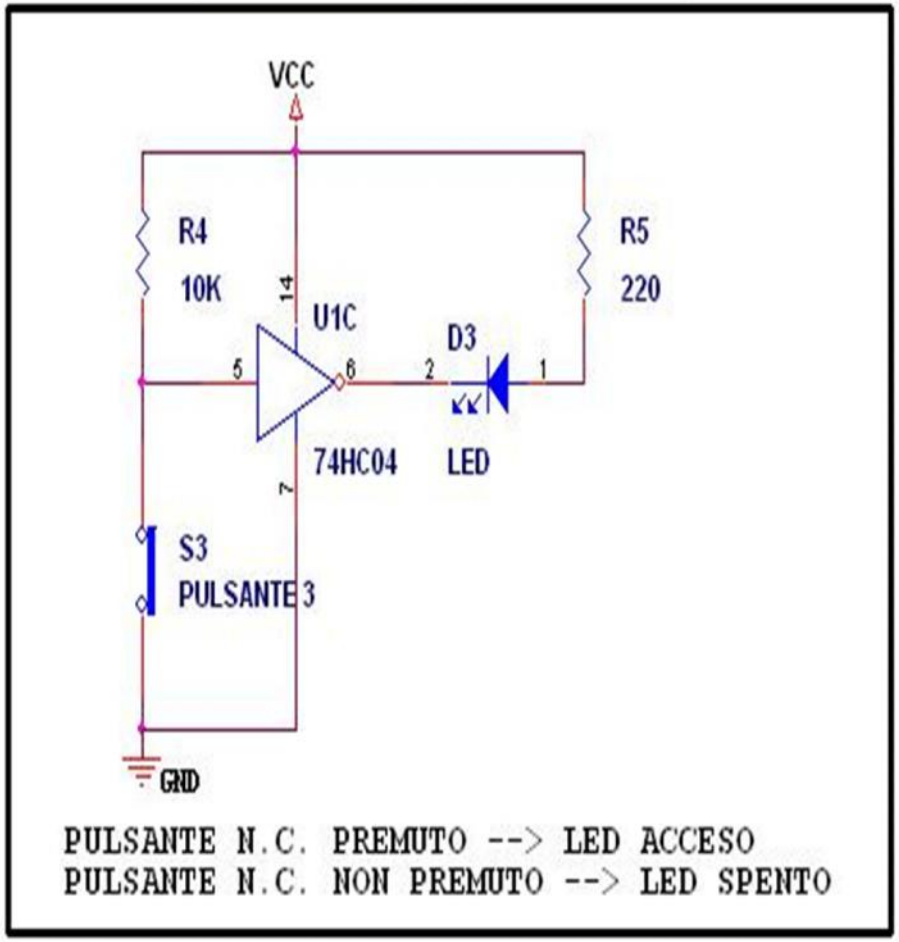
Gestione dei pulsanti con Arduino

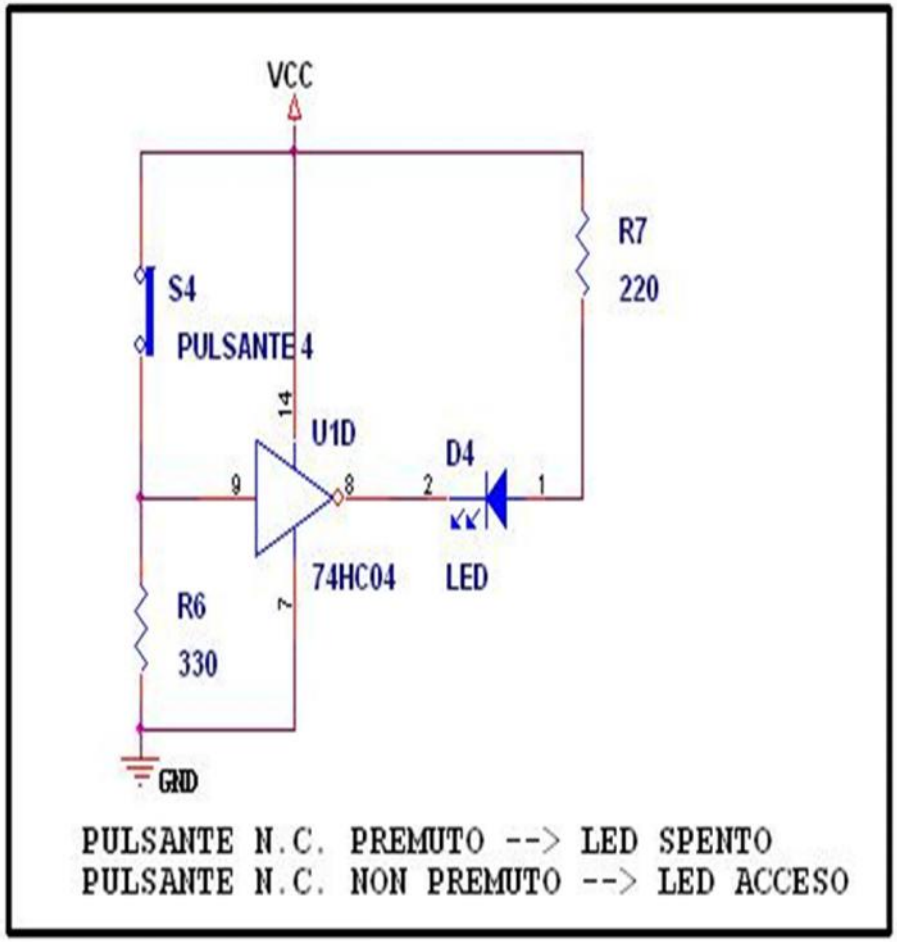




Circuiti di controllo di un pulsante n.a. senza l'utilizzo della scheda Arduino

Gestione dei pulsanti con Arduino

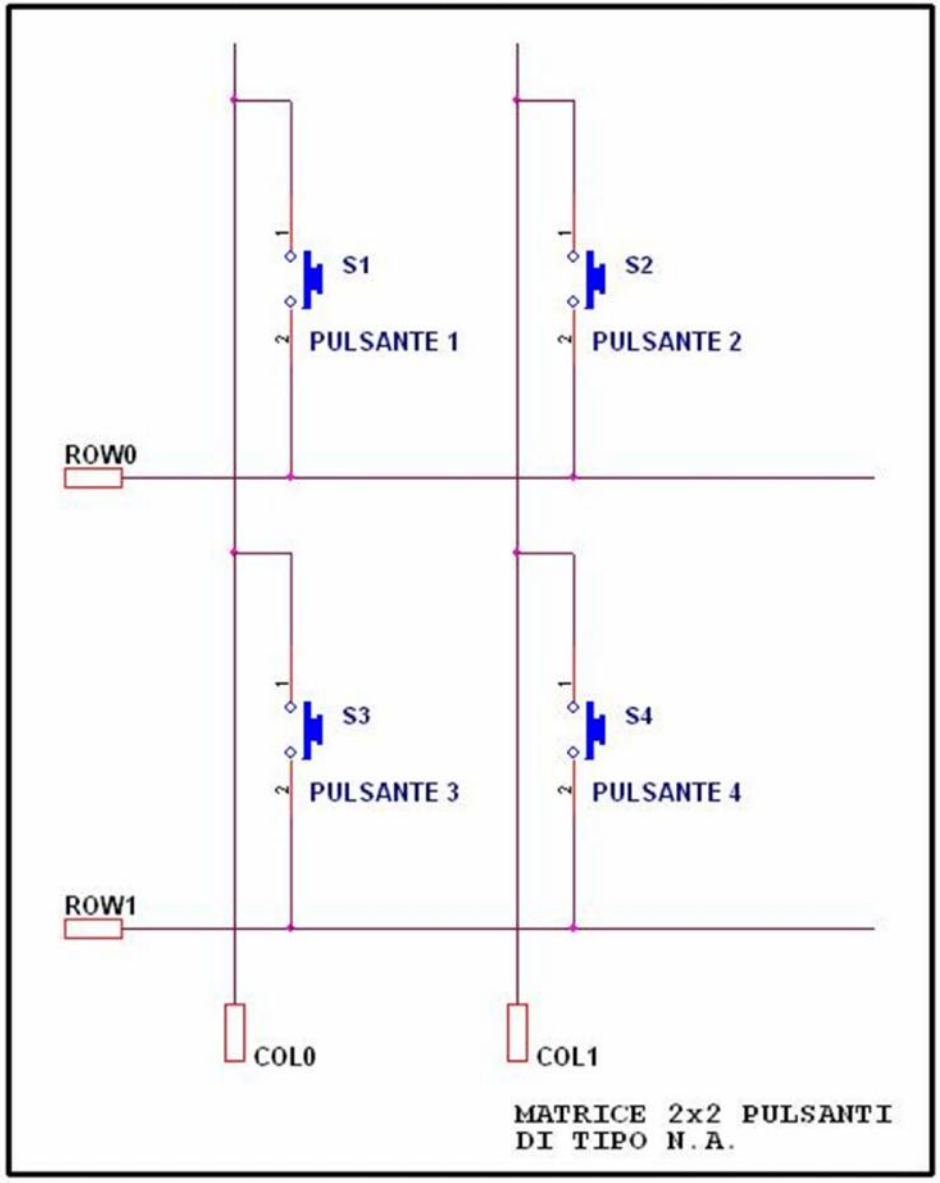


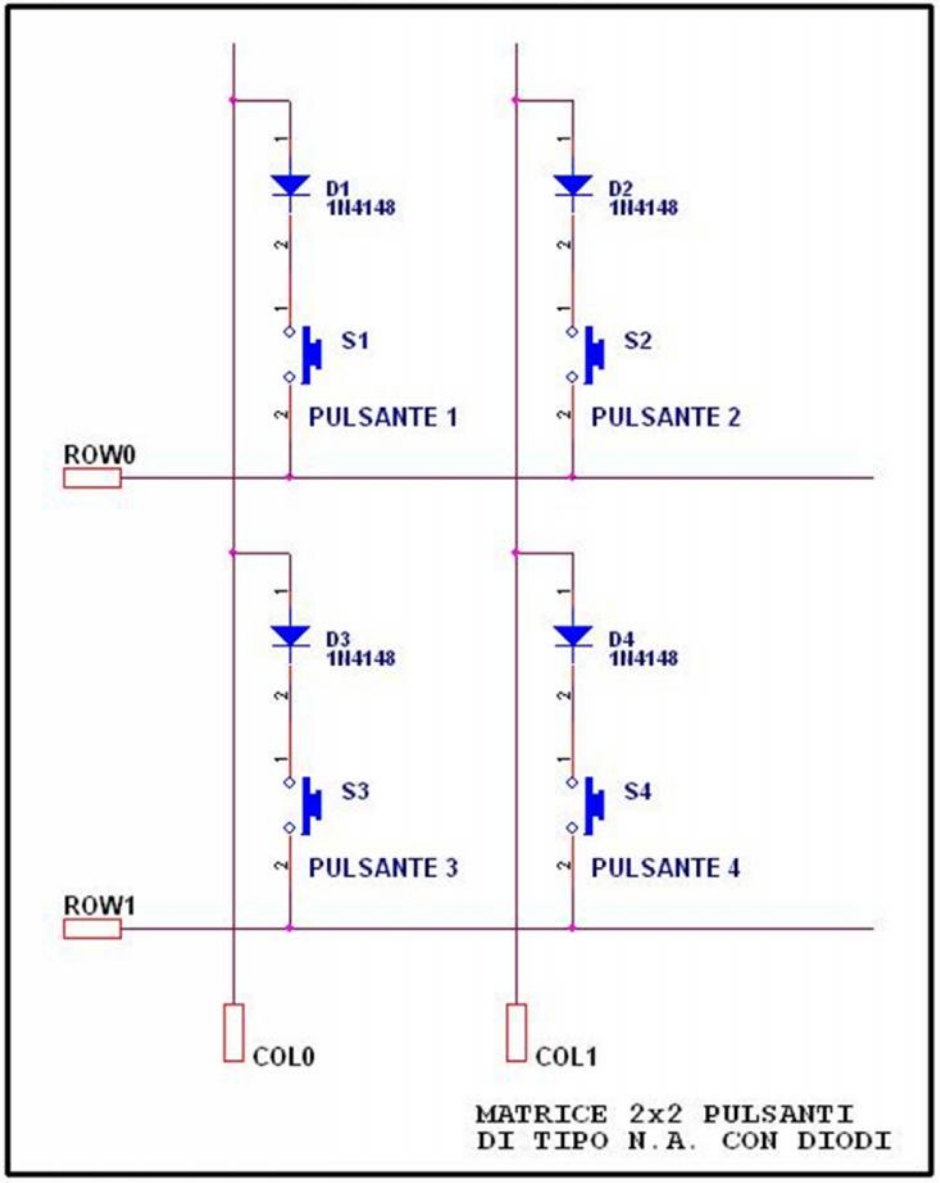


Circuiti di controllo di un pulsante n.c. senza l'utilizzo della scheda Arduino

Gestione dei pulsanti con Arduino

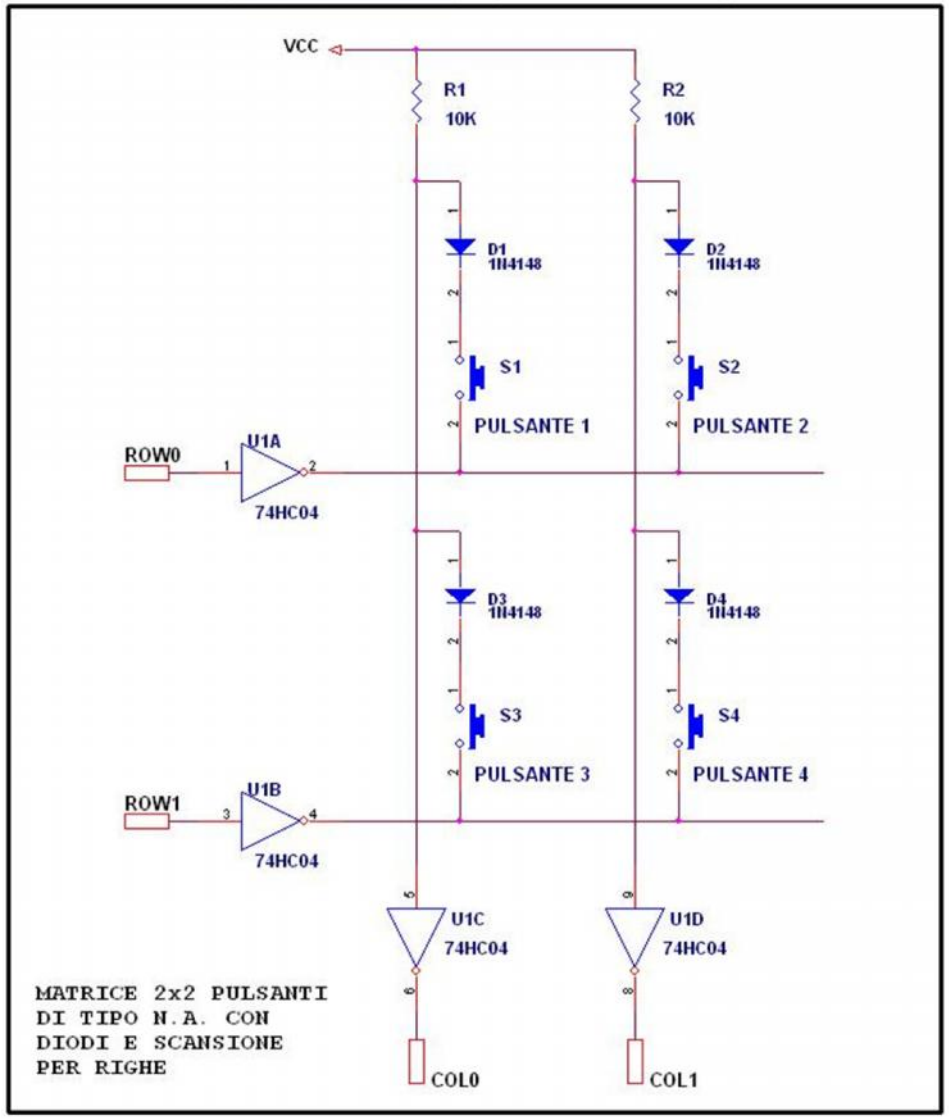
Keypad_matrix_2x2.ino

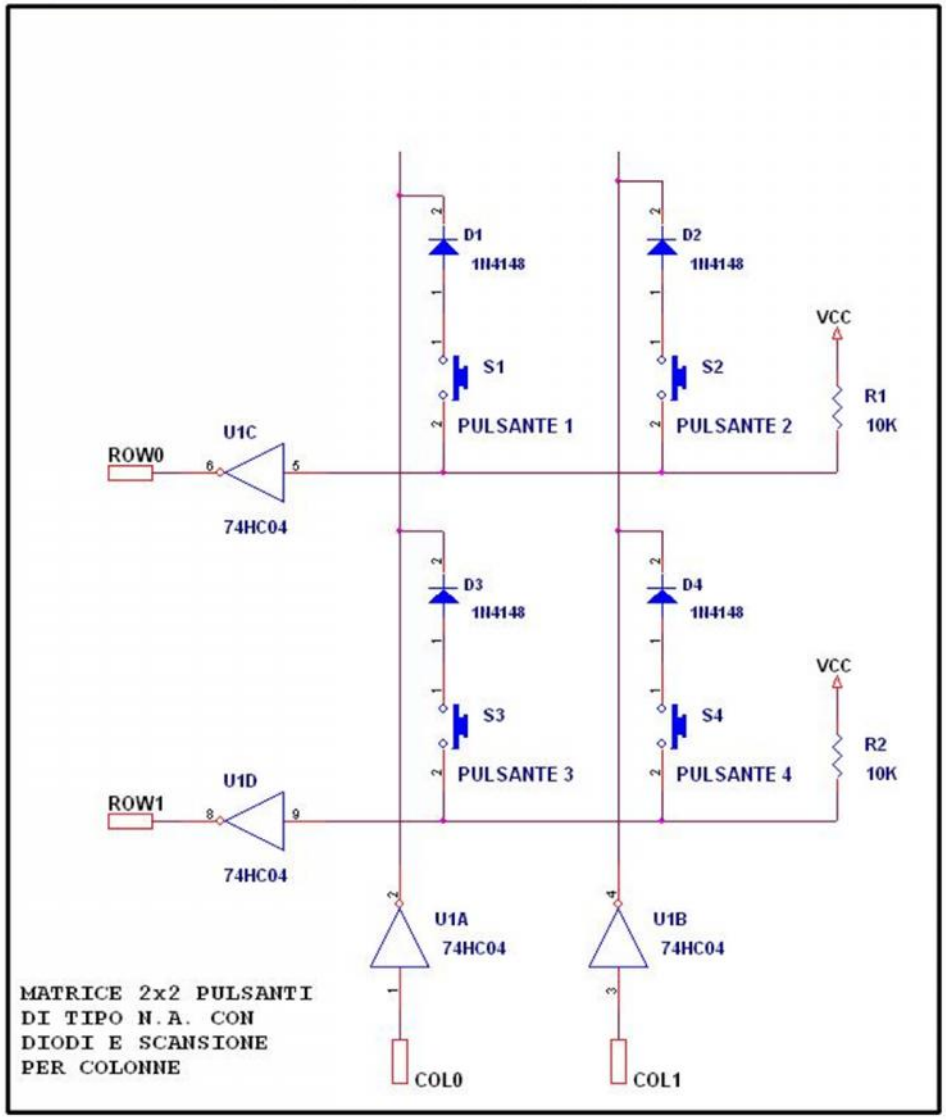




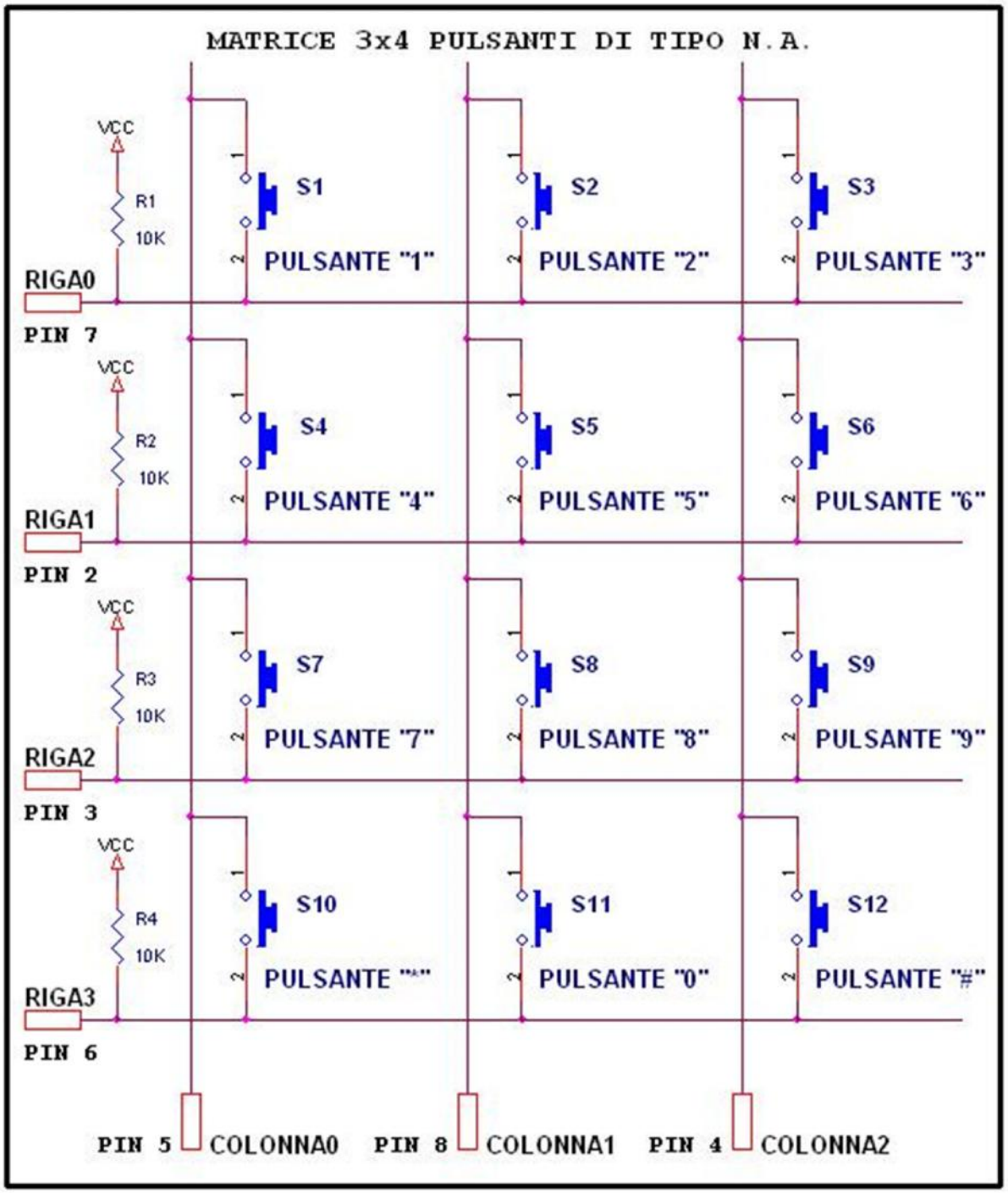
Schema elettrico di una matrice di pulsanti n.a. formata da 2 righe per 2 colonne senza diodi di protezione (fig. sinistra) e con diodi (fig. destra)

Gestione dei pulsanti con Arduino





Circuiti di controllo di una matrice di pulsanti n.a. formata da 2 righe per 2 colonne con l'utilizzo dei diodi di protezione



Gestione dei pulsanti con Arduino

Schema elettrico di una matrice di pulsanti n.a. formata da 4 righe per 3 colonne

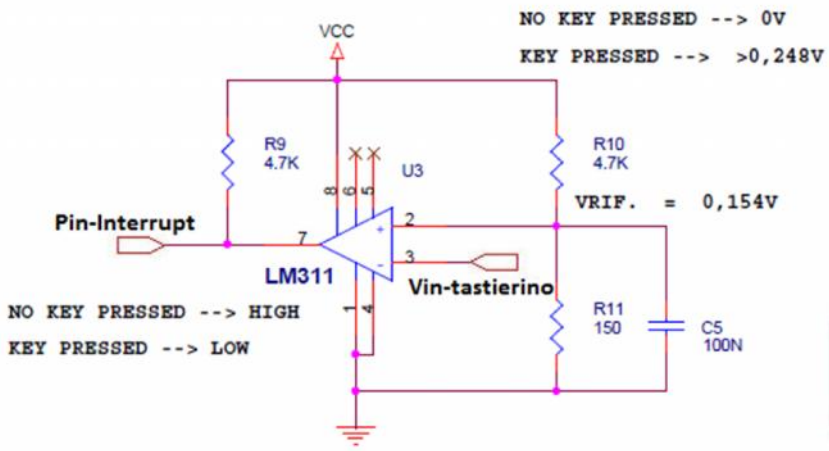
Keypad_3x4.ino

TASTO PREMUTO	Tensione in ingresso al ADC	Valore in decimale (10 bit)
1	0,248015873	50
2	0,433746425	88
3	0,641492266	131
4	0,968992248	198
5	1,52173913	311
6	2,020057307	413
7	2,688172043	550
8	3,395522388	695
9	3,831521739	784
*	4,237288136	867
0	4,55	931
#	4,7	962
NESSUNO	0	0

Gestione di una matrice di pulsanti

Keypad_3x4_analogico.ino

Schema elettrico di una matrice di pulsanti n.a. formata da 4 righe per 3 colonne gestita con un solo ingresso analogico della scheda Arduino



Fine lavori.
Per ora!