

# Semaforo Intelligente con Arduino

A cura del prof. Giuseppe SPALIERNO – 30/12/2011

In questa applicazione Arduino pilota i tre colori di un semaforo vincolando la durata del rosso all'intensità luminosa rilevata da una fotoresistenza.

La presenza di un'auto sulla fotoresistenza riduce la luminosità acquisita dal sensore e la durata del rosso, nell'esercitazione proposta, viene ridotta a 2s. anziché i normali 10s., come la durata del verde.

Prima di mettere a punto sia l'hardware da collegare ad Arduino che il software di gestione dell'applicazione in linguaggio C, si è disegnato il circuito da realizzare su breadboard grazie al software free "Fritzing" – [www.fritzing.org](http://www.fritzing.org) che consente di disegnare circuiti elettrici ed elettronici esteticamente molto accattivanti.

## ABOUT FRITZING

Fritzing is an open-source initiative to support designers, artists, researchers and hobbyists to work creatively with interactive electronics. We are creating a software and website in the spirit of [Processing](#) and [Arduino](#), developing a tool that allows users to **document** their prototypes, **share** them with others, **teach** electronics in a classroom, and to create a pcb layout for professional **manufacturing**.



## DOWNLOAD AND START!

[Download our latest version](#) (0.6.4b was released December 16th) and start right away.

Just got into interactive electronics and still need the basic tools? We created an "all-you-need-to-get-going" [Fritzing Starter Kit](#).



## PRODUCE YOUR BOARD

With the new [Fritzing Fab](#) service you can easily and inexpensively turn your sketch into a real, custom PCB. Try it out now!



## PARTICIPATE

Fritzing can only act as a creative platform if many people are using it as a means of sharing and learning. [Let us know](#) how it fits your needs and how it doesn't, [show it to your friends](#), and [share your projects](#).

## WHO'S BEHIND IT

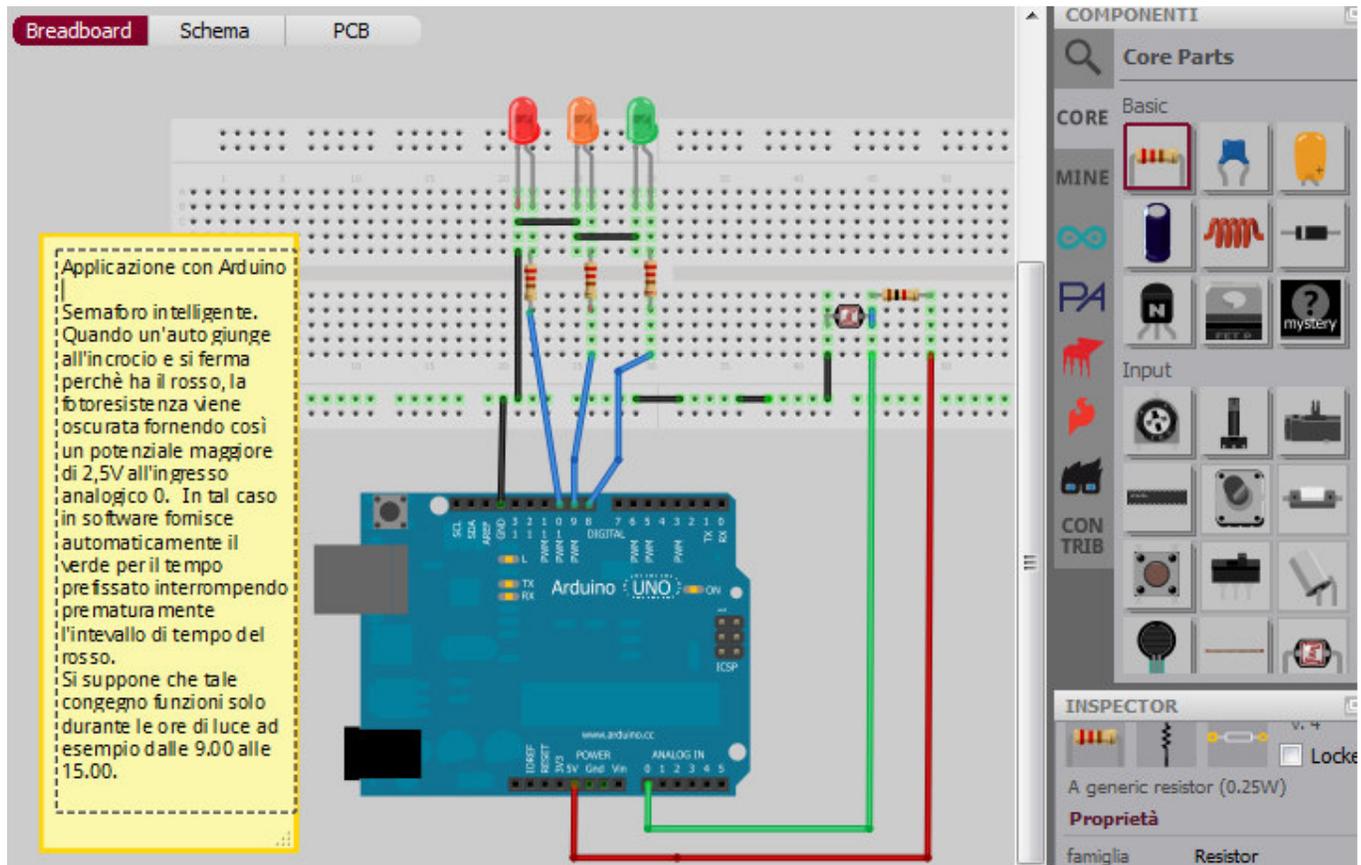
Fritzing is being developed by [researchers](#) in the [Interaction Design Lab](#) at the [University of Applied Sciences Potsdam](#),

Il software mette a disposizione una breadboard d una serie di componenti elettrici ed elettronici: dalle semplici resistenze, condensatori, diodi, transistor, interruttori, LED, display, sensori, circuiti integrati sino a moduli di notevole complessità come Arduino.

L'aspetto interessante è la possibilità di disegnare il circuito così come lo si realizza su breadboard, visualizzando la sagoma reale tridimensionale del componente oltre ai cavetti colorati di collegamento.

La costruzione del circuito è assai semplice: si preleva dalla libreria COMPONENTI l'elemento desiderato e lo si posiziona su breadboard. È possibile ruotare il componente selezionato e modificarne le caratteristiche, ad esempio il valore della resistenza, condensatore, colore del LED, ecc. In questa esercitazione si utilizzano, al posto delle lampade semaforiche, tre diodi LED di colore verde, giallo e rosso. Gli altri componenti utilizzati sono 3 resistenze da 220Ω per limitare l'intensità di corrente elettrica in ciascuno dei tre LED, una resistenza da 1KΩ da inserire in serie al sensore di luminosità impiegato che consiste in una fotoresistenza che presenta una resistenza di buio di 300KΩ ed una resistenza di 16KΩ a 10lux ed un valore inferiore a 1 KΩ in condizioni di forti luminosità.

Il software consente di inserire dei cavetti di collegamento tra i vari componenti ed è in grado di visualizzare il circuito non solo in modalità breadboard ma anche in "vista schema" ed in "vista PCB" per la realizzazione del circuito stampato.



Nel programma realizzato in Arduino 1.0 (scaricabile dal sito <http://arduino.cc/en/Main/Software>), in sintonia alle scelte hardware, si impostano le costanti Verde, Giallo e Rosso rispettivamente sulle porte DIGITAL 8, 9 e 10 e la costante Sonda sulla linea 0 del canale ANALOG IN.

Il blocco **void setup()** consente di impostare Verde, Giallo e Rosso come linee di uscita e Sonda come linea di ingresso. Il segnale analogico di ingresso deve essere contenuto nell'intervallo compreso tra 0 e +5V ed il codice corrispondente di uscita è un valore compreso tra 0 e 1023 poiché il convertitore A/D interno al microcontrollore presente sulla scheda Arduino è a 10bit ( $2^{10} = 1024$ ).

Viene, inoltre, inizializzata la porta seriale alla velocità di 9600bps (bit/s).

Il blocco **void loop()** pone sulle uscite Verde, Giallo e Rosso i valori, rispettivamente, Alto-Basso-Basso per 10s, Alto-Alto-Basso per 2s e Basso-Basso-Alto per 10s se il valore acquisito dalla fotoresistenza corrisponde ad un numero inferiore a 500 (condizione di luce) altrimenti per 2s. (condizione di buio).

```

Semaforo2 | Arduino 1.0
File Edit Sketch Tools Help
Semaforo2
// Semaforo intelligente
// La durata del rosso dipende dalla presenza di un'auto ferma
const int Verde=8;
const int Giallo=9;
const int Rosso=10;
const int Sonda=0;

void setup()
{
  pinMode(Verde, OUTPUT);
  pinMode(Giallo, OUTPUT);
  pinMode(Rosso, OUTPUT);
  pinMode(Sonda, INPUT);
}

void loop()
{
  int valore = analogRead(Sonda);
  if (valore < 500)
  {
    digitalWrite(Verde, HIGH);
    digitalWrite(Giallo, LOW);
    digitalWrite(Rosso, LOW);
    delay(10000);
  }
  else
  {
    digitalWrite(Verde, LOW);
    digitalWrite(Giallo, HIGH);
    digitalWrite(Rosso, LOW);
    delay(2000);
  }
  digitalWrite(Verde, LOW);
  digitalWrite(Giallo, LOW);
  digitalWrite(Rosso, HIGH);
  delay(10000);
}

```

Per compilare il programma e trasferirlo su Arduino si fa click sul pulsante freccia a destra che compare sotto la barra dei menù nella precedente figura.

Attivando la finestra “serial-monitor” (lente di ingrandimento a destra nella barra degli strumenti) sarà possibile visualizzare il valore acquisito durante ciascun ciclo grazie all’istruzione **Serial.println(val)** ove val è il codice numerico associato alla tensione acquisita sulla linea 0 di ANALOG IN.

Si riporta, di seguito, il listato del programma.

```
// Semaforo intelligente
// La durata del rosso dipende dalla presenza di un'auto ferma
const int Verde=8;
const int Giallo=9;
const int Rosso=10;
const int Sonda=0;

void setup()
{
  pinMode(Verde, OUTPUT);
  pinMode(Giallo, OUTPUT);
  pinMode(Rosso, OUTPUT);
  pinMode(Sonda, INPUT);
  Serial.begin(9600);
}

void loop()
{
  digitalWrite(Verde, HIGH);
  digitalWrite(Giallo, LOW);
  digitalWrite(Rosso, LOW);
  delay(10000);
  digitalWrite(Verde, HIGH);
  digitalWrite(Giallo, HIGH);
  digitalWrite(Rosso, LOW);
  delay(2000);
  digitalWrite(Verde, LOW);
  digitalWrite(Giallo, LOW);
  digitalWrite(Rosso, HIGH);
  int val=analogRead(Sonda);
  Serial.println(val);
  if (val<500) {
    delay(10000); }
  else {
    delay(2000); }
}
```