



**I.I.S. PRIMO LEVI**

**CLASSE 4C**

**PROGETTI REALIZZATI  
CON LA SCHEDA ARDUINO**



## Luci Natalizie

### Gruppo 1

#### Progettazione

É richiesto il controllo con Arduino dei led per simulare delle luci natalizie

#### Scopo dell'esercitazione

Apprendere il controllo del delay e il suo funzionamento che è in millisecondi

#### Codice

```
int led_pari[4];
int led_dispari[4];
void setup() {
    // put your setup code here, to run once:
    for(int i=0; i<4; i++){
        led_pari[i]=(i+1)*2;
        led_dispari[i]=((i+1)*2)+1;
        pinMode(led_pari[i],OUTPUT);
        pinMode(led_dispari[i],OUTPUT);
    }
}

void loop() {
    // put your main code here, to run repeatedly:
    for(int i=3; i>=0; i--){
        digitalWrite(led_dispari[i],HIGH);
        digitalWrite(led_pari[i],LOW);
    }

    delay(500);
    for(int i=3; i>=0; i--){
        digitalWrite(led_dispari[i],LOW);
        digitalWrite(led_pari[i],HIGH);
    }
    delay(500);
}
```



## Progetto Arduino: Led supercar

### Gruppo 2

#### Descrizione del progetto

Il progetto consisteva nell'accendere i led in sequenza una alla volta, e una volta che accesi tutti si procedeva ad accenderli in senso contrario

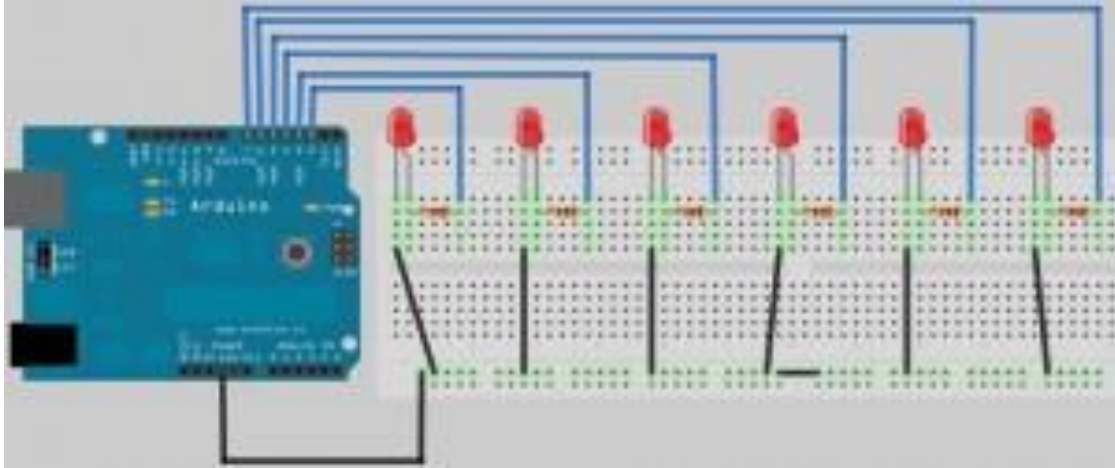
#### Codice Arduino

```
int led_yellow_01 = 9;    /dichiarazioni delle variabili dei led
int led_red_01 = 8;
int led_gre_01 = 7;
int led_blue_01 = 6;
int led_yellow_02 = 5;
int led_red_02 = 4;
int led_gre_02 = 3;
int led_blue_02 = 2;
int loadPin=0;          /variabile per lo stato del led
void setup() {
    pinMode(led_yellow_01, OUTPUT);    /dichiaro i valori dei led in input
    pinMode(led_red_01, OUTPUT);
    pinMode(led_gre_01, OUTPUT);
    pinMode(led_blue_01, OUTPUT);
    pinMode(led_yellow_02, OUTPUT);
    pinMode(led_red_02, OUTPUT);
    pinMode(led_gre_02, OUTPUT);
    pinMode(led_blue_02, OUTPUT);
}

void loop() {                /ciclo loop di arduino
    for (loadPin=2; loadPin<10; loadPin++)    /ciclo for per accendere i led in ordine crescente
    {
        digitalWrite(loadPin,HIGH);        /accensione del led
        delay(100);                        /tempo di accensione
        digitalWrite(loadPin,LOW);         /spegnimento del led
    }
    for (loadPin=9; loadPin>1; loadPin--)    /ciclo per accendere i led in ordine decrescente
    {
        digitalWrite(loadPin,HIGH);        /accensione del led
        delay(100);                        /tempo di accensione
        digitalWrite(loadPin,LOW);         /spegnimento del led
    }
}
```



**Schema elettrico**





## Semaforo con pulsante

### Gruppo 3

#### Scopo dell'esercitazione:

Attivare un sistema semaforico attraverso l'utilizzo di un bottone, in modo tale che il programma si avvii dopo aver premuto il pulsante, che si accenda un led alla volta con la sequenza prevista (rosso, verde, giallo).

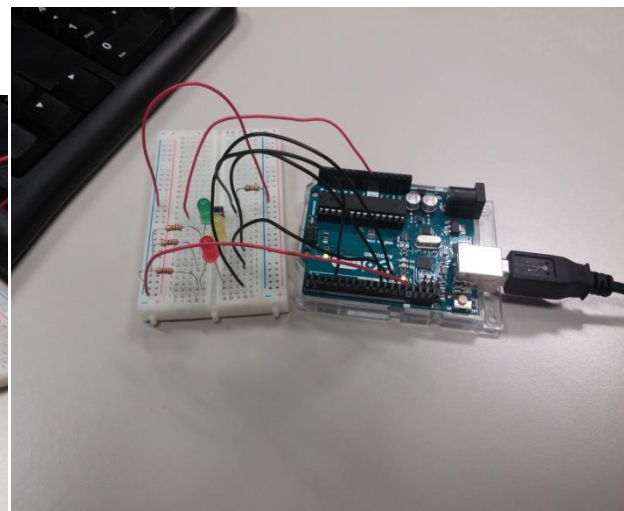
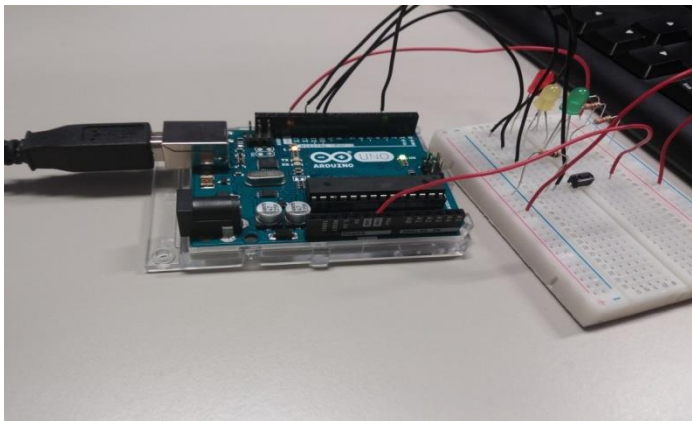
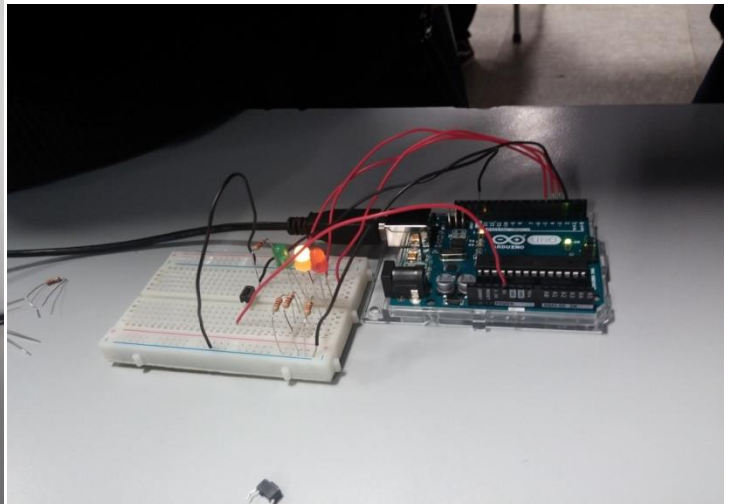
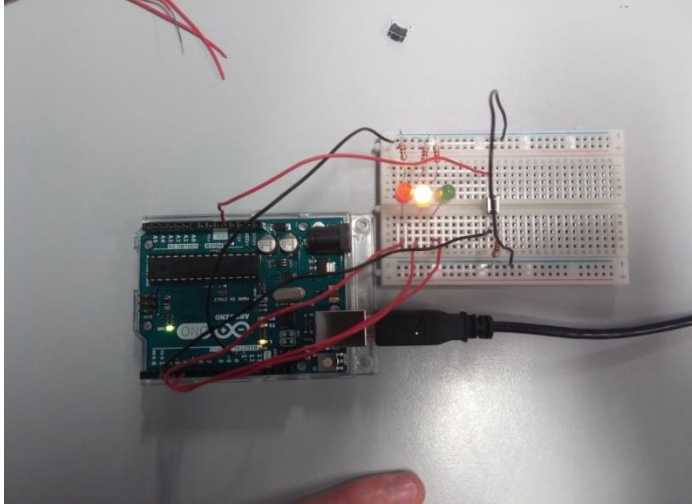
#### Codice

```
int rosso=10;
int verde=11;
int giallo=12;
int button=4;
void setup() {
    // put your setup code here, to run once:
    pinMode(rosso,OUTPUT);
    pinMode(verde,OUTPUT);
    pinMode(giallo,OUTPUT);
    pinMode(button,INPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    int read_button, i;
    read_button=digitalRead(button);
    if (read_button==LOW)
    {
        for(i=20;i>0;i--)
        {
            digitalWrite(rosso,HIGH);
            digitalWrite(verde,HIGH);
            digitalWrite(giallo,HIGH);
            delay(100);
            digitalWrite(rosso,LOW);
            digitalWrite(verde,LOW);
            digitalWrite(giallo,LOW);
            delay(100);
        }
    }
    else
    {
        digitalWrite(rosso,HIGH);
        delay(100);
        digitalWrite(rosso,LOW);
        delay(100);
    }
}
```



Immagini relative all'esercitazione:





# Due Semafori con due pulsanti di chiamata per il rosso

## Gruppo 4

Siccome dovevamo realizzare un progetto semaforico per la gestione del traffico AGV, abbiamo realizzato un progetto con questo fine ma in scala minore, per verificare se funzionasse correttamente. Per realizzarlo abbiamo usato la piattaforma Arduino.

## Codice

```
int rosso_1=12;
int giallo_1=11;
int verde_1=10;
int rosso_2=7;
int giallo_2=6;
int verde_2=5;
int bottone_1=3;
int bottone_2=2;

void setup() {
  // put your setup code here, to run once:
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(3,INPUT);
  pinMode(2,INPUT);
  digitalWrite(rosso_2,HIGH);
}

void loop() {
  // put your main code here, to run repeatedly:
  int i, button;
  digitalWrite(rosso_1,HIGH);
  digitalWrite(giallo_1,LOW);
  delay(3000);
  digitalWrite(rosso_2,LOW);
  digitalWrite(verde_2,HIGH);
  for(i=7000;i>0;i--){
    delay(1);
    button=digitalRead(bottone_2);
    if(button==LOW)
      i=0;
  }
  digitalWrite(verde_2,LOW);
  digitalWrite(giallo_2,HIGH);
  delay(3000);
  digitalWrite(giallo_2,LOW);
  digitalWrite(rosso_2,HIGH);
  delay(2000);
  digitalWrite(rosso_1,LOW);
  digitalWrite(verde_1,HIGH);

  for(i=7000;i>0;i--){
    delay(1);
    button=digitalRead(bottone_1);
    if(button==LOW)
      i=0;
  }
  digitalWrite(verde_1,LOW);

  digitalWrite(giallo_1,HIGH);
  delay(3000);
}
```



**Commento al codice:**

- 1) dichiarazione delle variabili
- 2) impostare i led come output e i bottoni come input (nel void setup) impostando il rosso2 come high.
- 3) abbiamo acceso il led rosso1 e verde2 e lasciato spento il led giallo1 e rosso2 (nel void loop)
- 4) abbiamo creato un ciclo for con un contatore chiamato "i" impostato come valore iniziale i=7000 che serve a verificare se il bottone viene premuto.
- 5) dopo aver fatto il ciclo abbiamo fatto accendere e spegnere i led per il primo semaforo
- 6) ripetere lo stesso ciclo per il secondo semaforo





## ***Titolo esperienza: Gestione di un led attraverso un bottone***

### **Gruppo 5**

#### ***Parte Sperimentale :***

Il programma funziona nel seguente modo: tramite una variabile, a ogni ciclo viene controllato se il led è già acceso o no, se il led risulta spento e il bottone risulta premuto, il led si accenderà. Se al contrario il led risulta spento, una volta premuto il bottone il led si accenderà.

#### **Codice:**

```
int led=2, button=10;
bool state=false, clicked=false;
void setup() {
  pinMode(led,OUTPUT);
  pinMode(button,INPUT);
}

void loop() {
  if((digitalRead(button)==LOW) && (state==false)){
    digitalWrite(led,HIGH);
    state=true;
    clicked=true;
  }
  else{
    if((digitalRead(button)==LOW) && (state==true)){
      digitalWrite(led,LOW);
      state=false;
      clicked=true;
    }
  }
  while(clicked==true){
    if(digitalRead(button)==HIGH)
      clicked=false;
  }
}
```



## Relazione TPS Arduino per FCA - Quiz

### Gruppo 6

#### Scopo dell'esercitazione

Lo scopo di questa esercitazione è di realizzare un circuito che regoli una pulsantiera per un gioco a quiz con 3 giocatori, ogni giocatore ha un pulsante, e il primo che preme il bottone ha il diritto di rispondere, dopodiché gli altri due pulsanti rimangono inattivi.

#### Codice

```
int lettura;
int count = 0;

//player1
int player1=3;
int btn_player1=2;

//player2
int player2=6;
int btn_player2=5;

//player3
int player3=9;
int btn_player3=8;

void setup() {
  //OUTPUT
  pinMode(player1,OUTPUT);
  pinMode(player2,OUTPUT);
  pinMode(player3,OUTPUT);

  //INPUT
  pinMode(btn_player1,INPUT);
  pinMode(btn_player2,INPUT);
  pinMode(btn_player3,INPUT);
}

void loop() {
  lettura=4*digitalRead(btn_player1)+2*digitalRead(btn_player2)+1*digitalRead(btn_player3);
  if(lettura==4 && count==0) {
    digitalWrite(player1,HIGH);
    digitalWrite(player2,LOW);
    digitalWrite(player3,LOW);
    count=1;
  }
  else
  {
    if(lettura==2 && count==0) {
      digitalWrite(player1,LOW);
      digitalWrite(player2,HIGH);
      digitalWrite(player3,LOW);
      count=1;
    }
    else
    {
      if(lettura==1 && count==0) {
```



```
digitalWrite(player1,LOW);  
digitalWrite(player2,LOW);  
digitalWrite(player3,HIGH);  
count=1;  
    }  
}  
}
```

Immagine del circuito

