

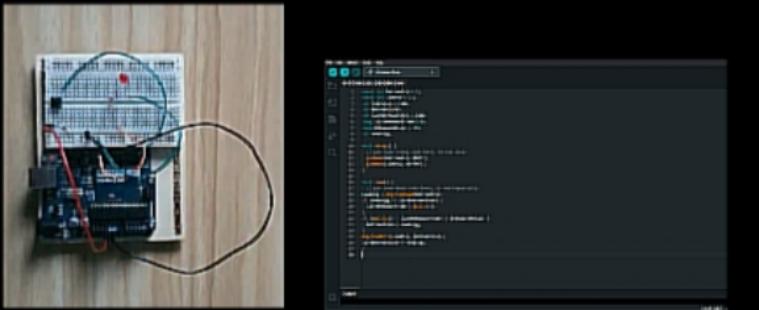
LAB 2 UTILIZZO MONITOR SERIALE E ARDUINO IDE

START

DI KARROUM ABDERRAHIM, PARMA CHRISTIAN
E WARNAKULASOORIYA RICCARDO FERNANDO

Riassunto

IN QUESTA PRESENTAZIONE
SPIEGHEREMO COME
REALIZZARE VARI CIRCUITI
UTILIZZANDO ARDUINO IDE PER
SCRIVERE DEI CODICI



```
const int ledPin = 13;
const int buttonPin = 2;
const int sensorPin = 0;
const int motorPin = 9;

void setup() {
  // Set the digital pins as output
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(sensorPin, INPUT);
  pinMode(motorPin, OUTPUT);

  // Set the analog pins as input
  analogReadResolution(10);
}

void loop() {
  // Read the value from the button
  if (digitalRead(buttonPin) == HIGH) {
    // Turn on the LED
    digitalWrite(ledPin, HIGH);
    // Turn off the motor
    digitalWrite(motorPin, LOW);
  } else {
    // Turn off the LED
    digitalWrite(ledPin, LOW);
    // Turn on the motor
    digitalWrite(motorPin, HIGH);
  }

  // Read the value from the sensor
  int sensorValue = analogRead(sensorPin);
  // Print the sensor value to the serial monitor
  Serial.println(sensorValue);
}
```

►CONTENUTI

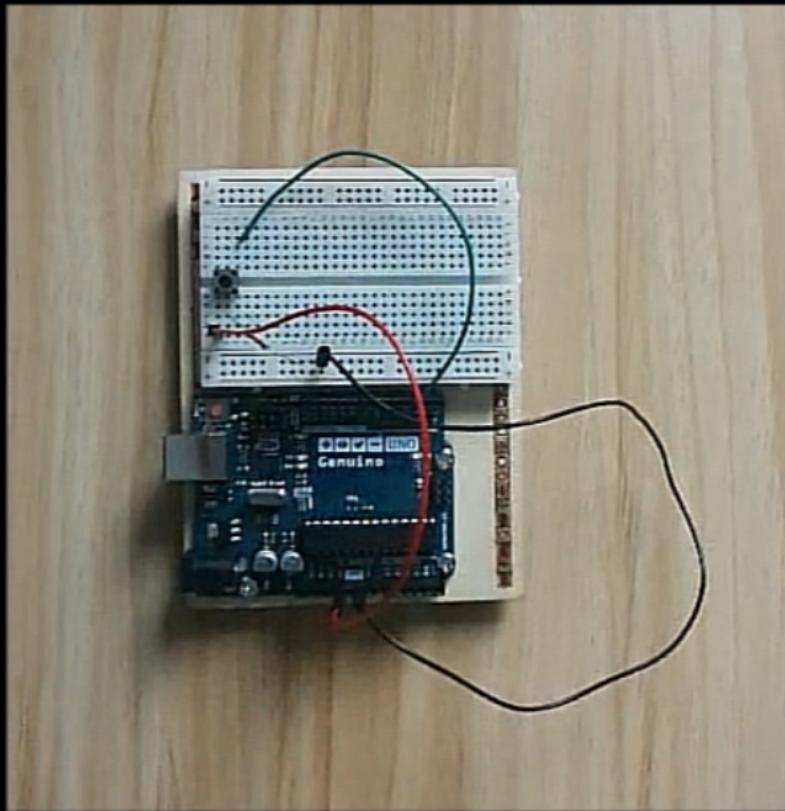
01. CIRCUITO 1

02. CIRCUITO 2

03. CIRCUITO 3

04. CIRCUITO 4

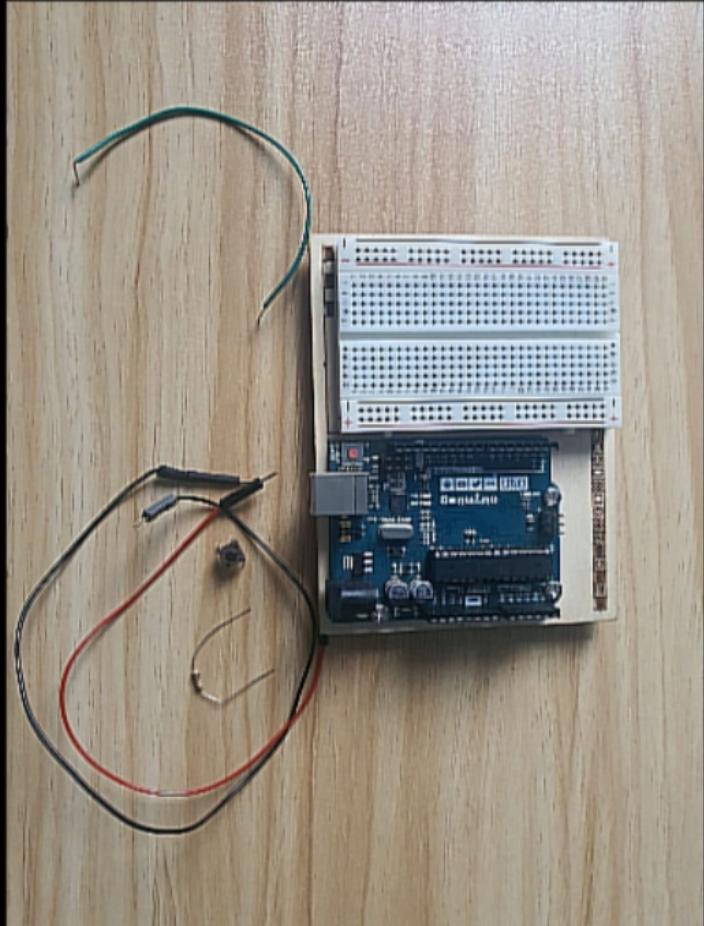
05. CIRCUITO 5



CIRCUITO 1

In questo circuito vi spiegheremo come utilizzare Arduino IDE per programmare il pushbutton, facendo apparire nel monitor seriale 0, se il bottone non è premuto, o 1, quando viene premuto.

+INFO



CIRCUITO 1

COMPONENTI

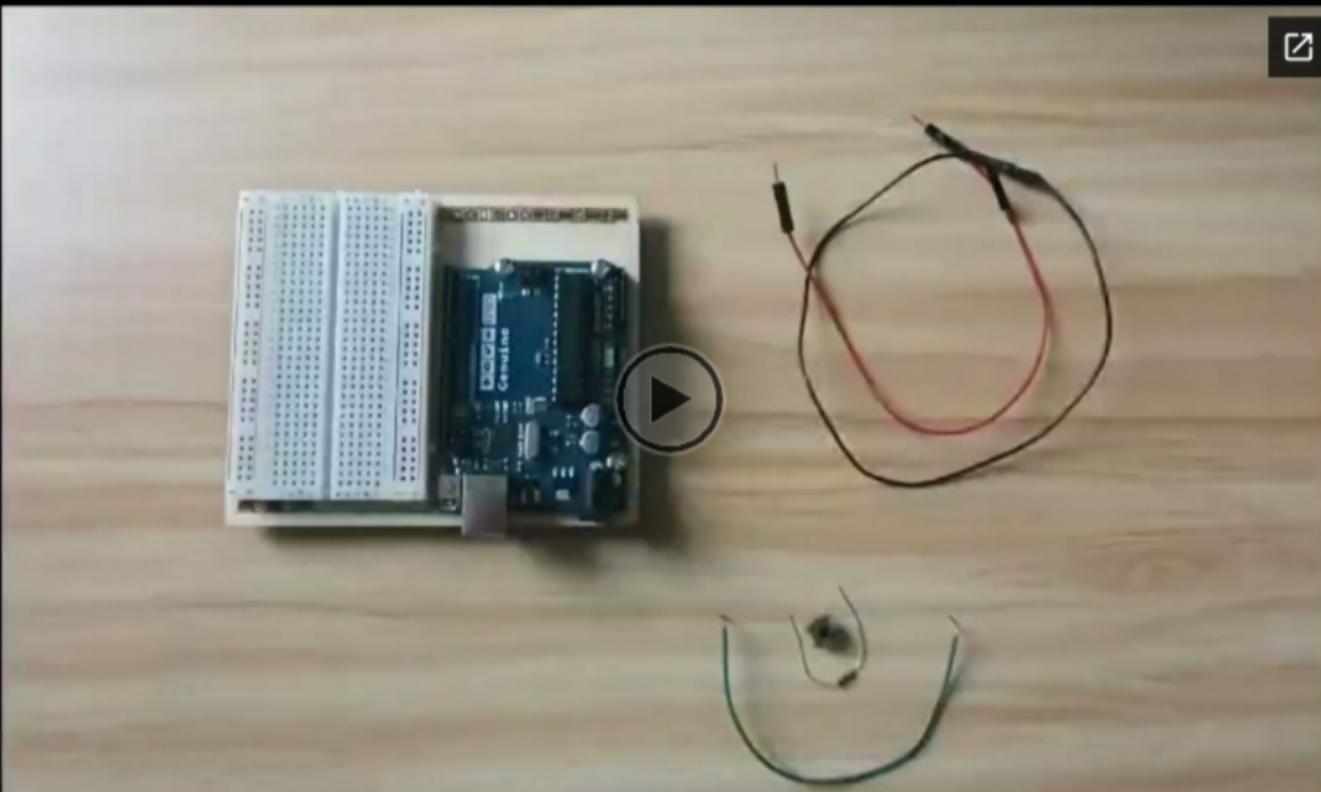
Per realizzare questo circuito,
serviranno:

- Scheda Arduino Uno;
- Breadboard;
- 2x Cavi Jumper;
- Resistenza da $10\text{k}\Omega$;
- Pushbutton;
- Ponticello

+INFO

CIRCUITO 1

MONTAGGIO



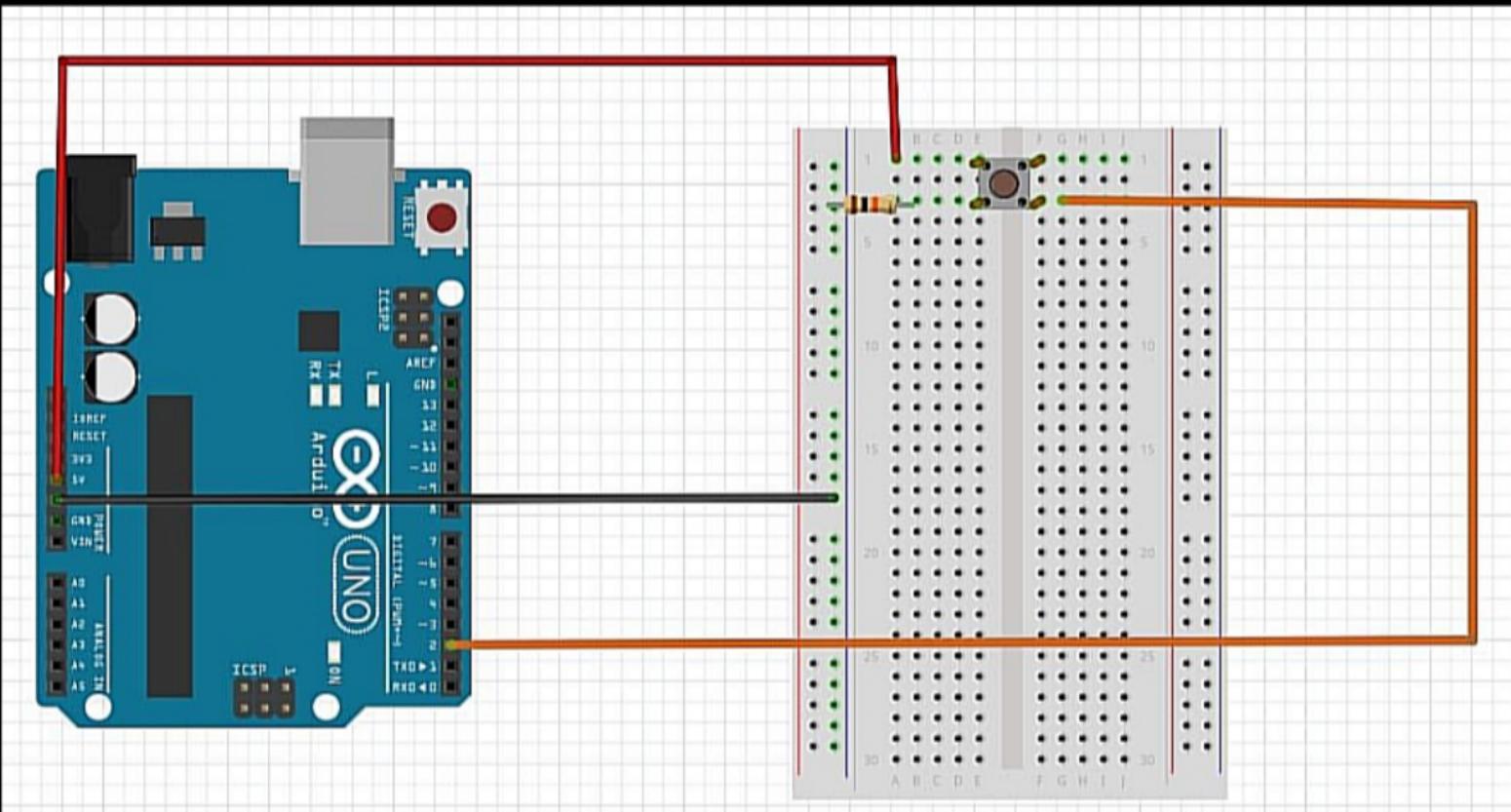
CIRCUITO 1

ACCENSIONE



CIRCUITO 1

SCHEMA FRITZING



CIRCUITO 1

CODICE

The screenshot shows the Arduino IDE interface with the following details:

- File Bar:** File, Edit, Sketch, Tools, Help.
- Sketch Name:** BOTTONE_MONITOR.SERIALE.0.1.ino
- Board:** Arduino Uno
- Code Area:** The main code area displays the following C++ code for a button monitoring sketch:

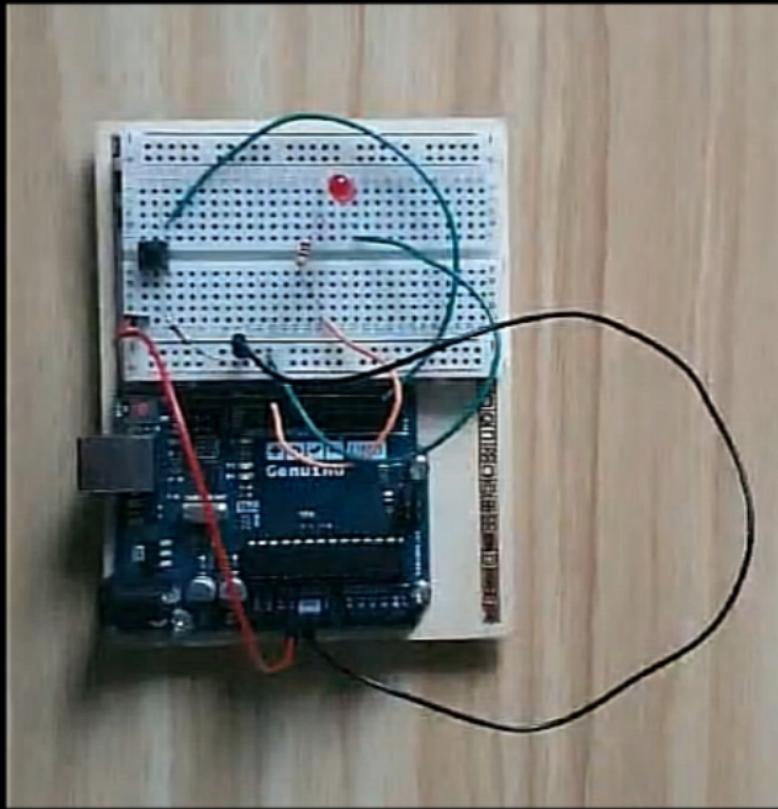
```
const int buttonPin = 2;
int buttonState;
int lastbuttonstate = LOW;
long lastDebounceTime = 0;
long DebounceDelay = 50;

void setup() {
    // put your setup code here, to run once:
    pinMode(buttonPin,INPUT);
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
    int buttonState = digitalRead(buttonPin);
    if(buttonstate != lastbuttonstate) {
        lastDebounceTime = millis();
    }
    if((millis() - lastDebounceTime) > DebounceDelay) {
        buttonState =! buttonState;
    }
    Serial.println(buttonState);
    lastbuttonstate = buttonState;
}
```

Output Tab: An Output tab is visible at the bottom left.

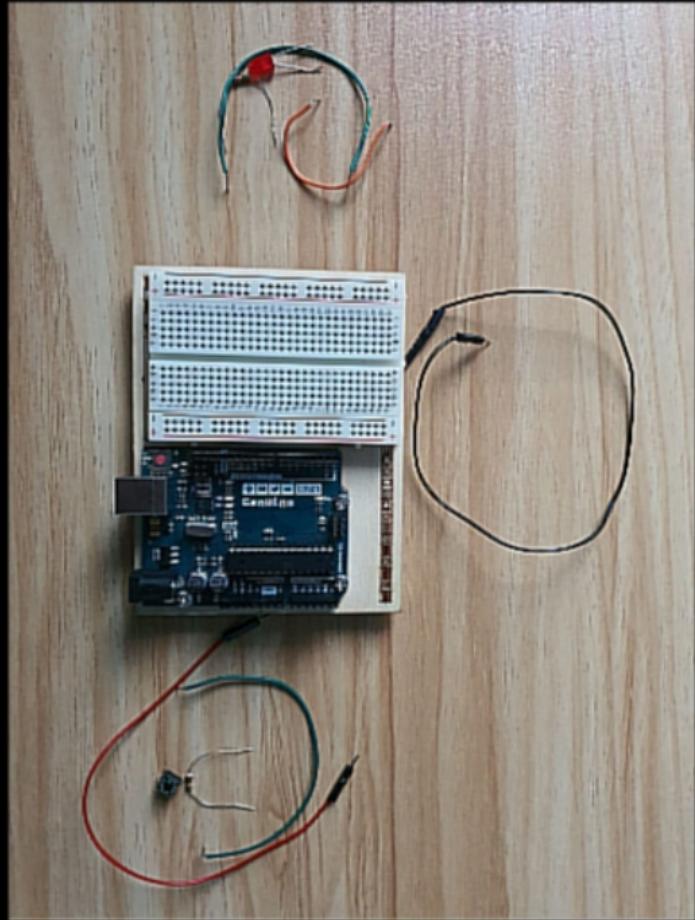
Status Bar: In 29, Col 1 | Arduino Uno on COM4 (not connected)



CIRCUITO 2

In questo circuito vi spiegheremo come, utilizzando Arduino IDE, si può accendere un led, grazie anche ad un bottone.

+INFO



CIRCUITO 2

COMPONENTI

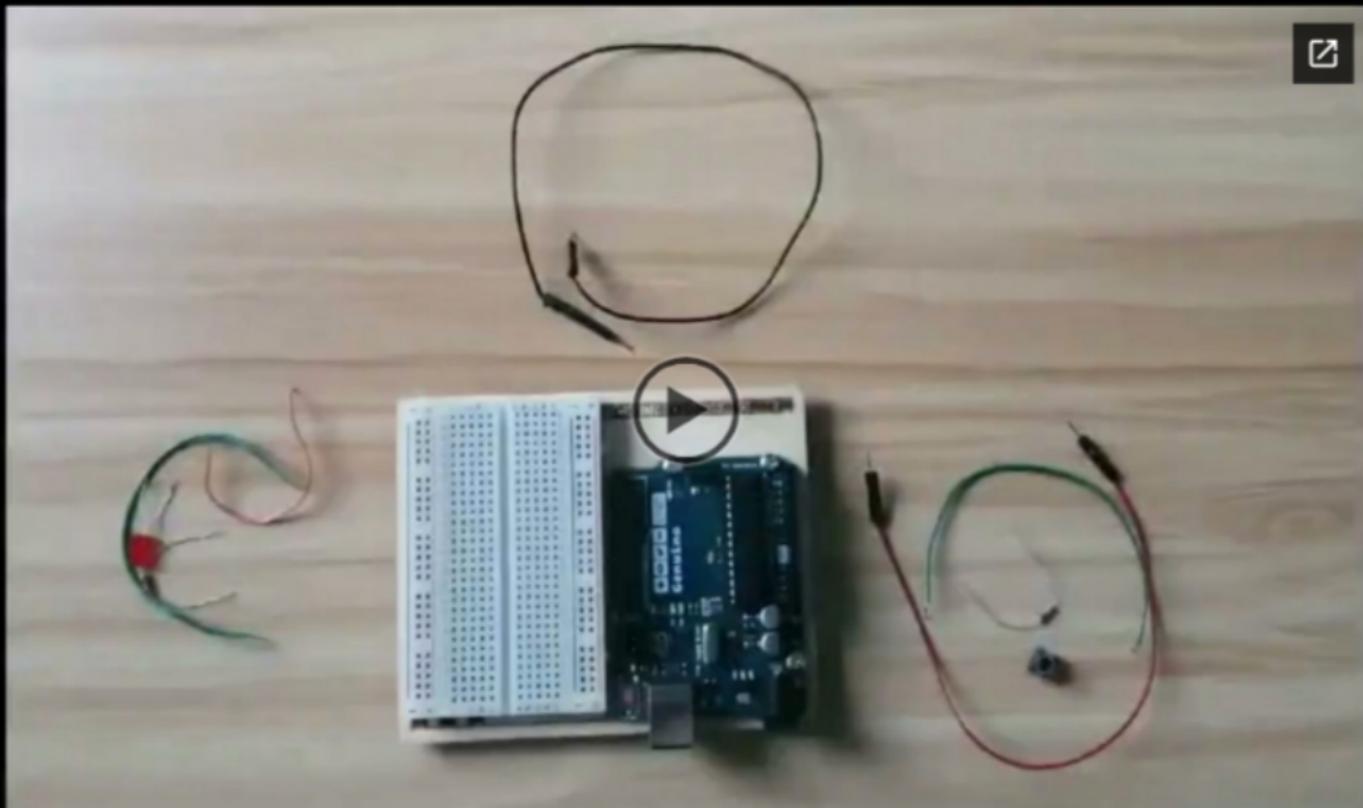
Per realizzare questo circuito,
serviranno:

- Scheda Arduino Uno;
- Breadboard;
- 2x Cavi Jumper;
- Resistenza da $10\text{k}\Omega$;
- Pushbutton;
- Resistenza da 220Ω ;
- 3x Ponticelli

+INFO

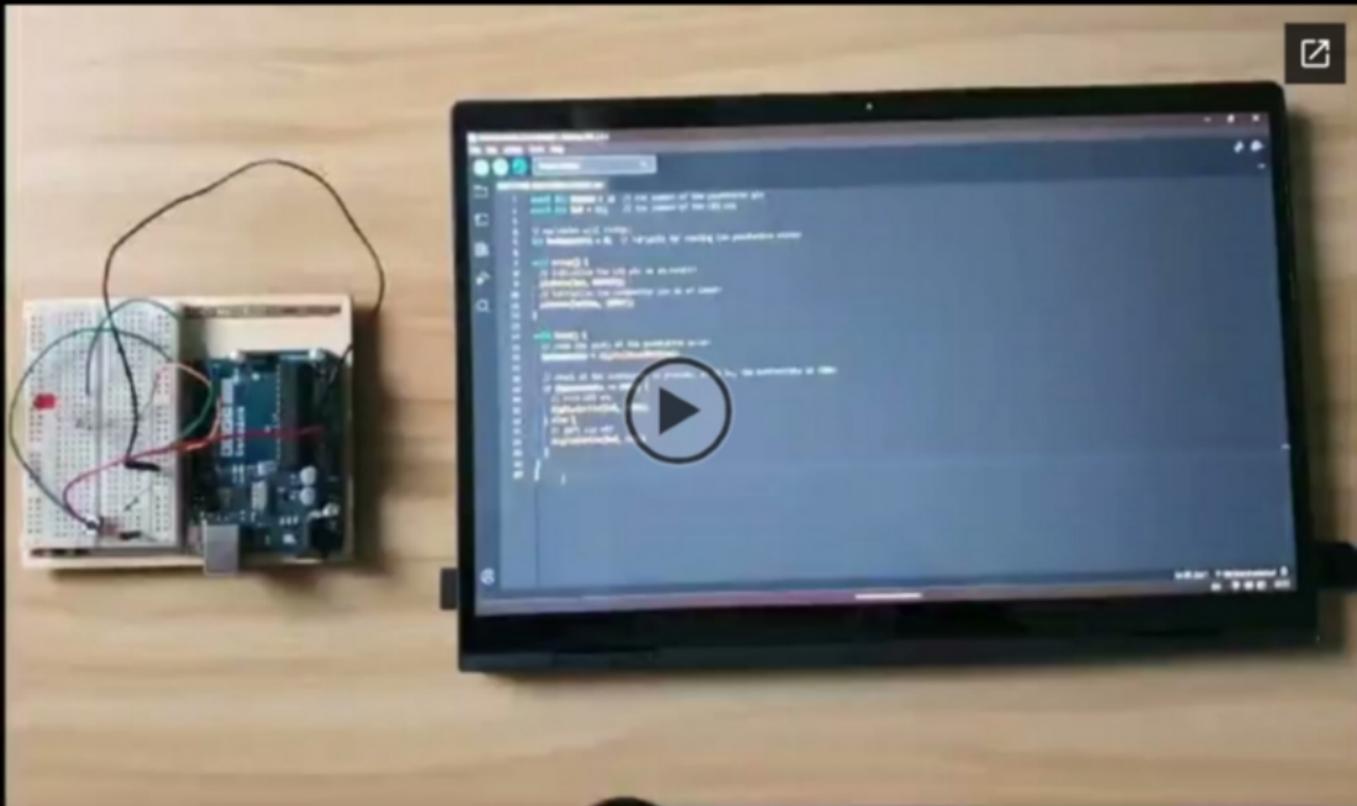
CIRCUITO 2

MONTAGGIO



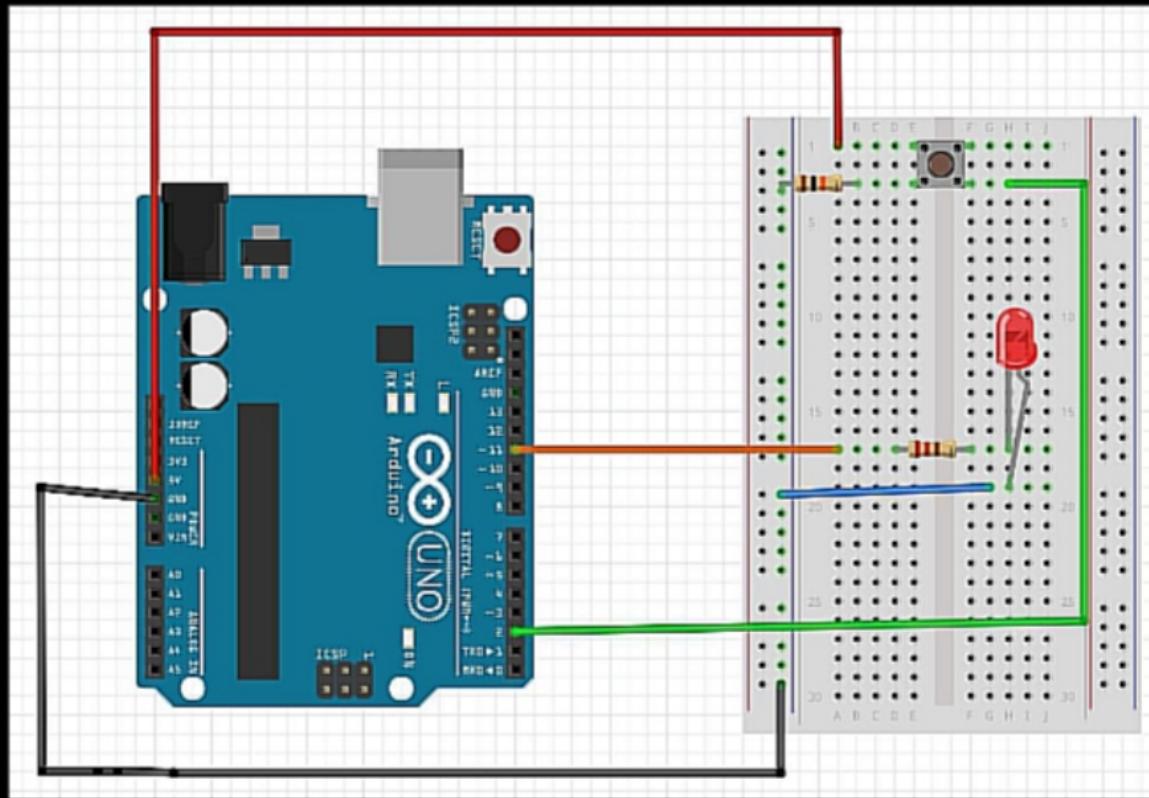
CIRCUITO 2

ACCENSIONE



CIRCUITO 2

SCHEMA FRITZING



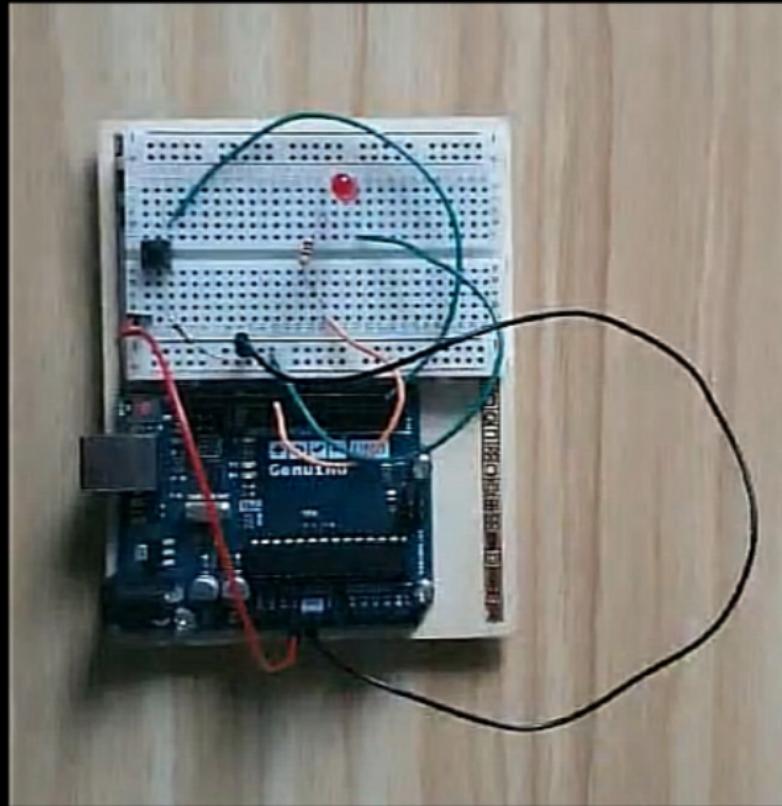
CIRCUITO 2

CODICE

The screenshot shows the Arduino IDE interface with the following details:

- File menu:** File, Edit, Sketch, Tools, Help.
- Toolbars:** Standard toolbar with icons for Open, Save, Print, and others.
- Arduino Uno selected in the top bar:** Shows the board type and connection status.
- Code Editor:** Displays the sketch named "BOTONE.LED.CODICE.ino".
- Code Content:**

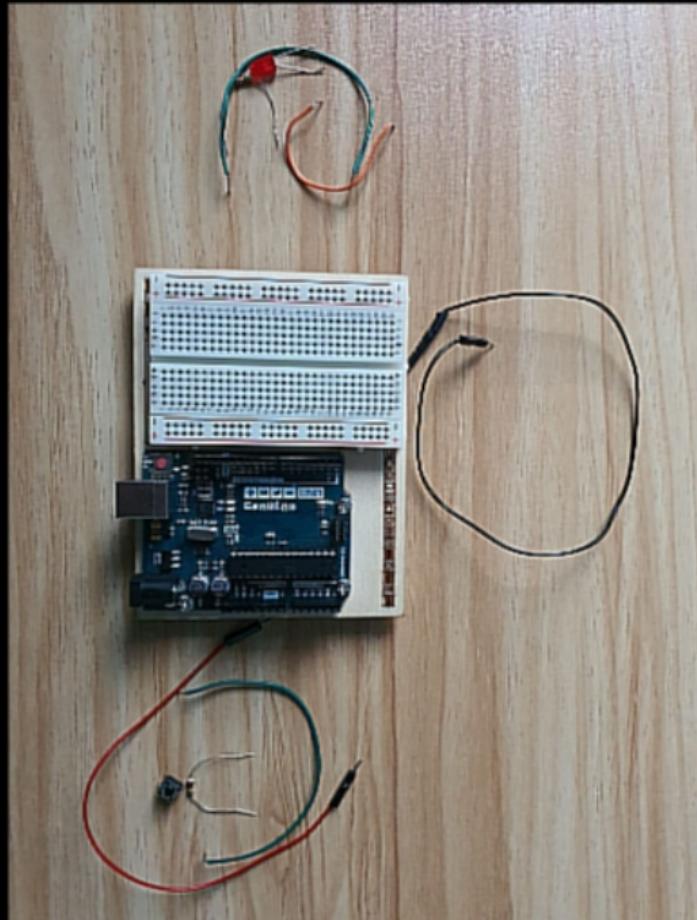
```
1 const int button = 2; // the number of the pushbutton pin
2 const int led = 11; // the number of the LED pin
3
4 // variables will change:
5 int buttonState = 0; // variable for reading the pushbutton status
6
7 void setup() {
8     // initialize the LED pin as an output:
9     pinMode(led, OUTPUT);
10    // initialize the pushbutton pin as an input:
11    pinMode(button, INPUT);
12 }
13
14 void loop() {
15     // read the state of the pushbutton value:
16     buttonState = digitalRead(button);
17
18     // check if the pushbutton is pressed, if it is, the buttonState is HIGH:
19     if (buttonState == HIGH) {
20         // turn LED on:
21         digitalWrite(led, HIGH);
22     } else {
23         // turn LED off:
24         digitalWrite(led, LOW);
25     }
26 }
27 }
```
- Output Tab:** Shows the output "In 27. Col 1 - Arduino Uno on COM4 last connected!"



CIRCUITO 3

In questo circuito vi spiegheremo come accendere un diodo LED utilizzando un pushbutton e, grazie ad Arduino IDE, aggiungeremo un debouncer.

+INFO



CIRCUITO 3

COMPONENTI

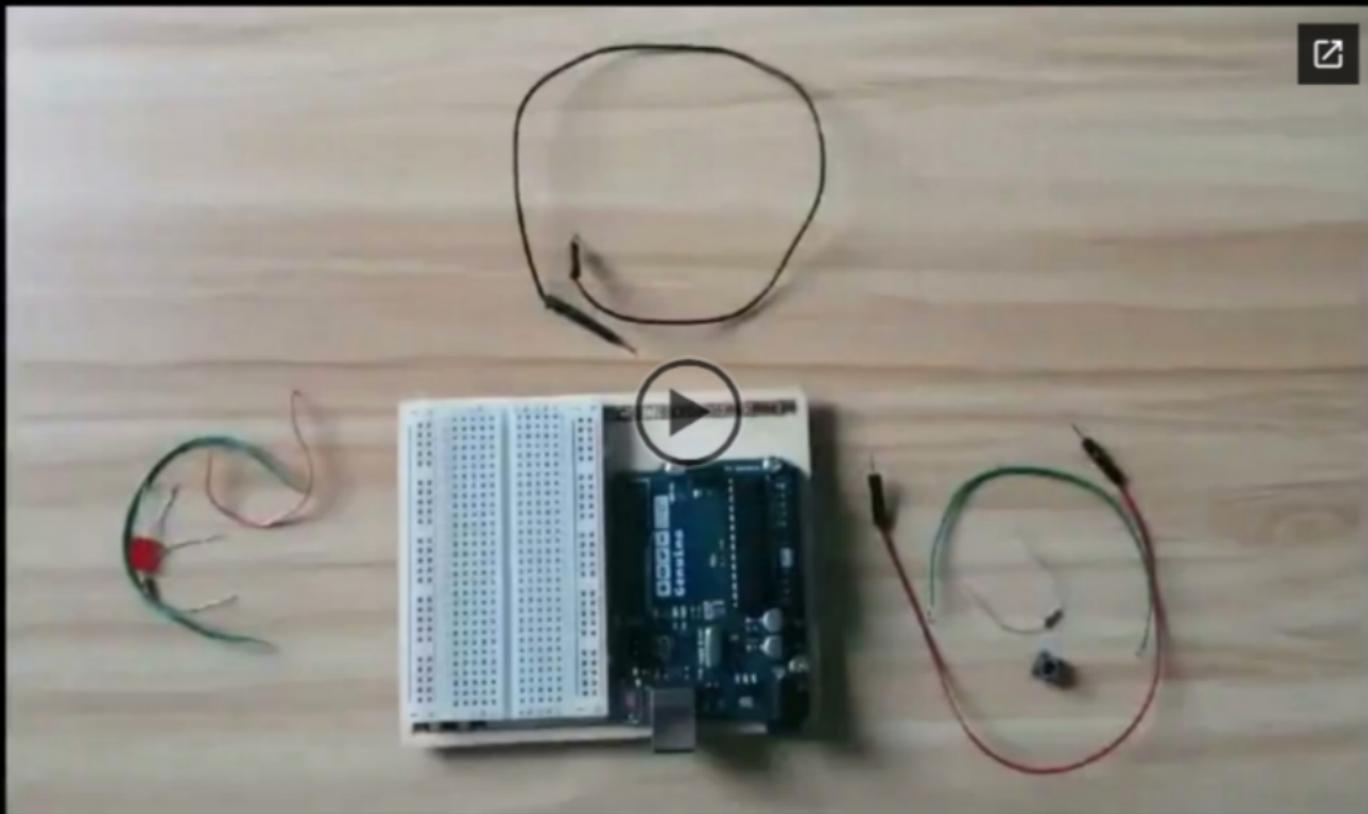
Per realizzare questo circuito,
serviranno:

- Scheda Arduino Uno;
- Breadboard;
- 2x Cavi Jumper;
- Resistenza da $10\text{k}\Omega$;
- Pushbutton;
- Resistenza da 220Ω ;
- 3x Ponticelli

+INFO

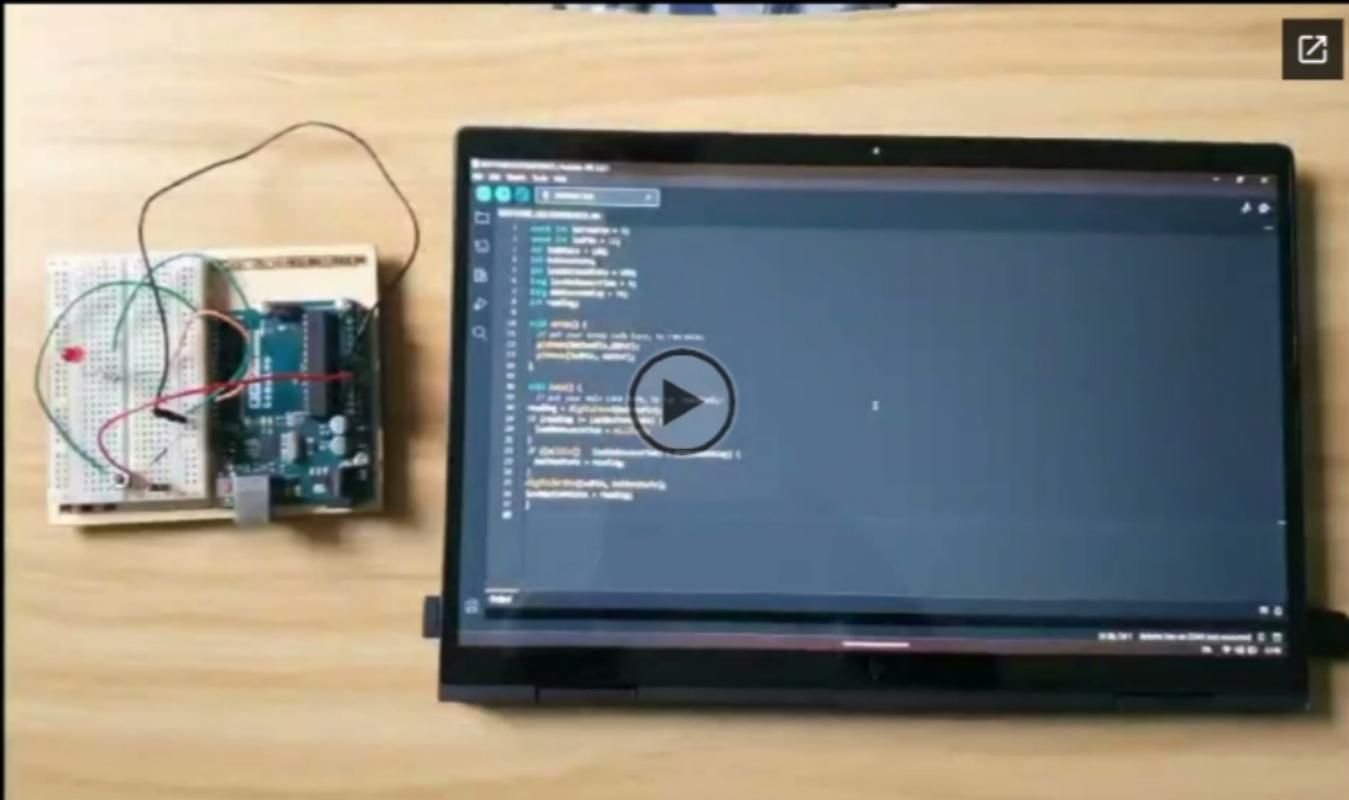
CIRCUITO 3

MONTAGGIO



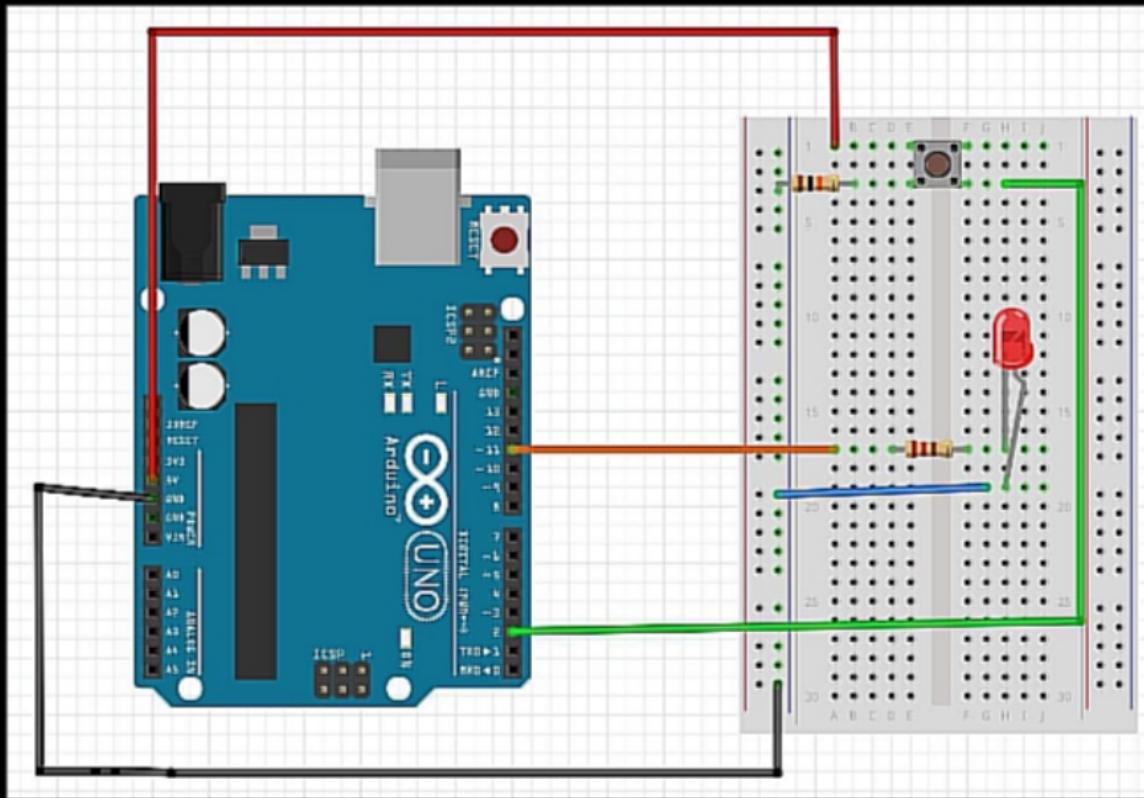
CIRCUITO 3

ACCENSIONE



CIRCUITO 3

SCHEMA FRITZING



CIRCUITO 3

CODICE

The screenshot shows the Arduino IDE interface with the following details:

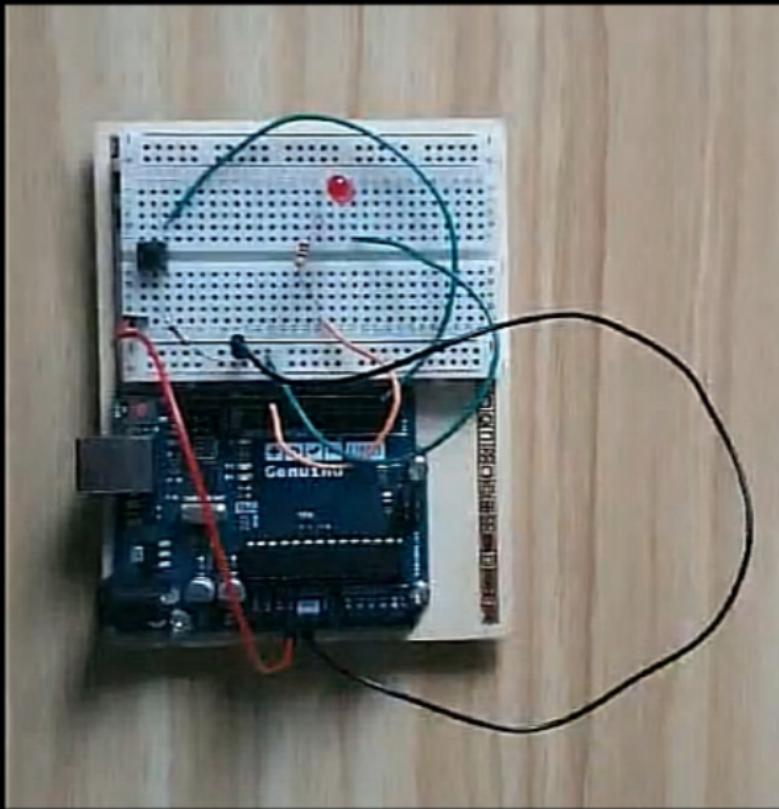
- File menu:** File, Edit, Sketch, Tools, Help.
- Sketch menu:** Sketch, Tools, Help.
- Board menu:** Arduino Uno.
- Code area:** A dark-themed code editor containing the following C++ code for a debouncing circuit:

```
const int buttonPin = 2;
const int ledPin = 13;
int ledState = LOW;
int buttonState;
int lastButtonState = LOW;
long lastDebounceTime = 0;
long debounceDelay = 50;
int reading;

void setup() {
    // put your setup code here, to run once:
    pinMode(buttonPin, INPUT);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    reading = digitalRead(buttonPin);
    if (reading != lastButtonState) {
        lastDebounceTime = millis();
    }
    if ((millis() - lastDebounceTime) > debounceDelay) {
        buttonState = reading;
    }
    digitalWrite(ledPin, buttonState);
    lastButtonState = reading;
}
```

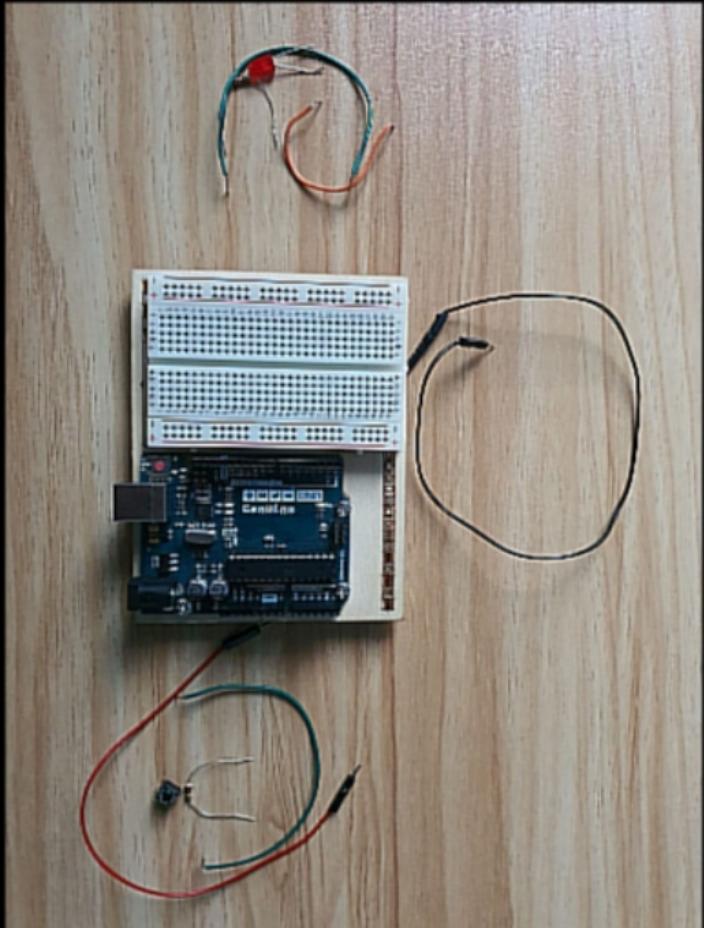
Output tab: Shows the message "In 28, Col 1 Arduino Uno on COM4 (not connected)".



CIRCUITO 4

In questo circuito vi spiegheremo come accendere un diodo LED e farlo rimanere acceso con un debouncer, inoltre il monitor seriale dirà 0 se il LED è spento e 1 se il LED è acceso.

+INFO



CIRCUITO 4

COMPONENTI

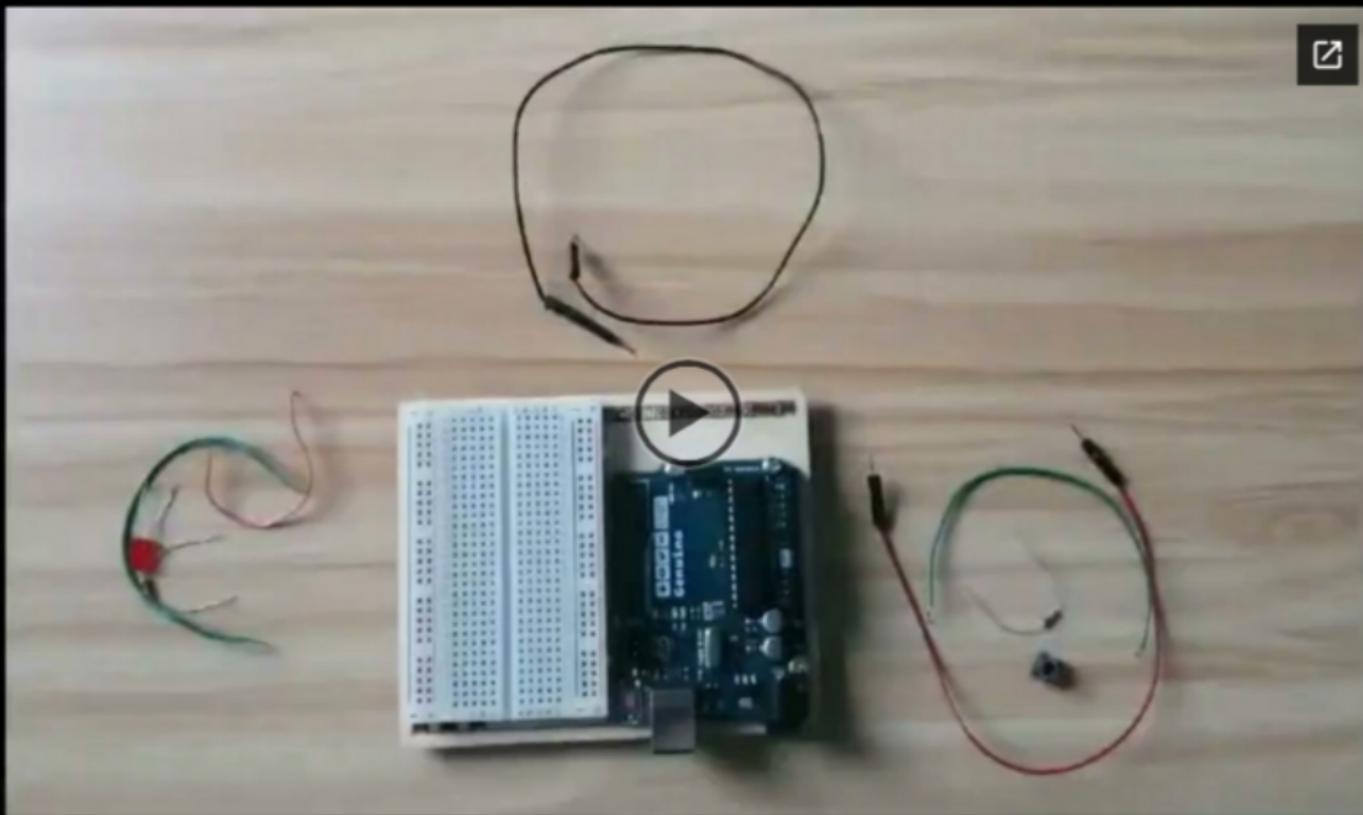
Per realizzare questo circuito,
serviranno:

- Scheda Arduino Uno;
- Breadboard;
- 2x Cavi Jumper;
- Resistenza da $10\text{k}\Omega$;
- Pushbutton;
- Resistenza da 220Ω ;
- 3x Ponticelli

+INFO

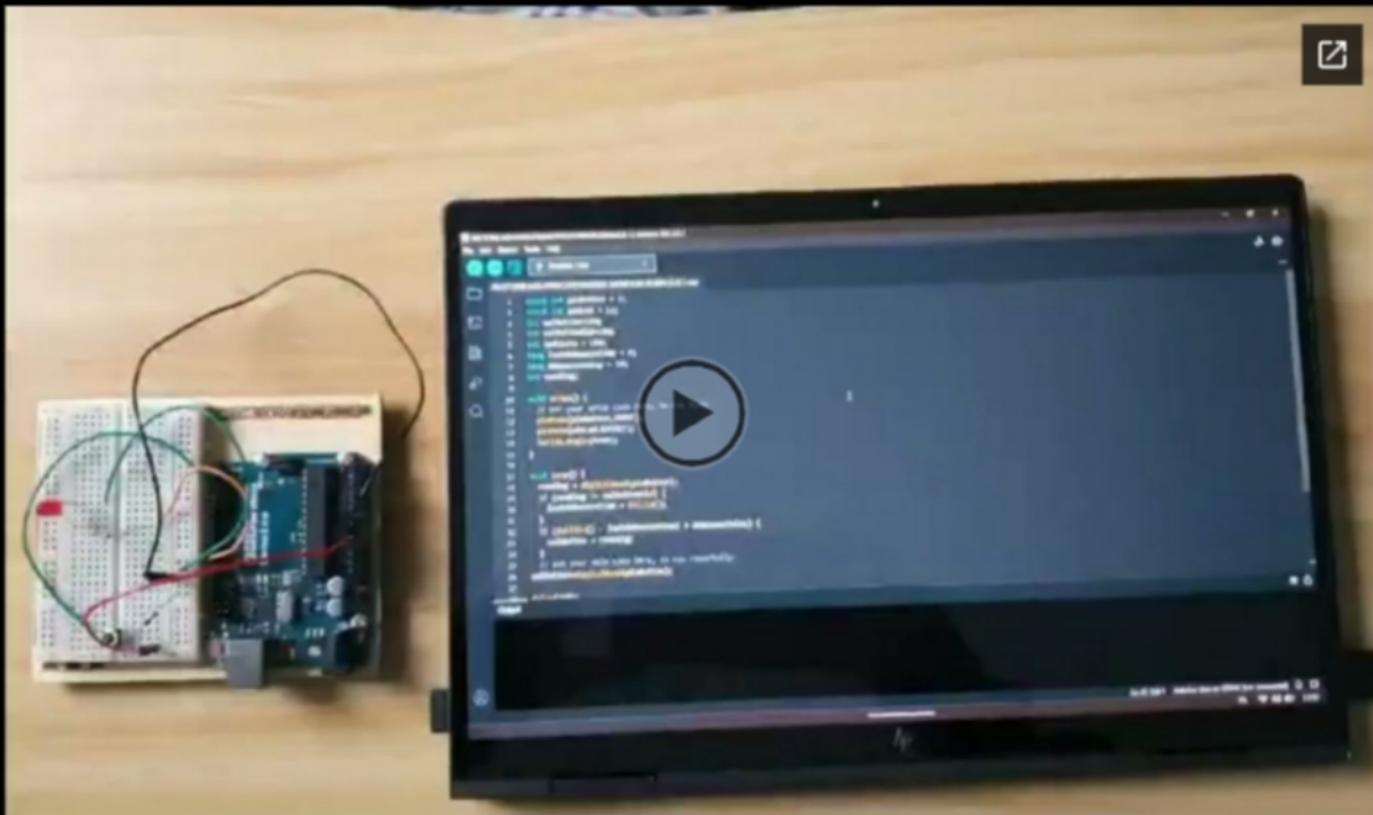
CIRCUITO 4

MONTAGGIO



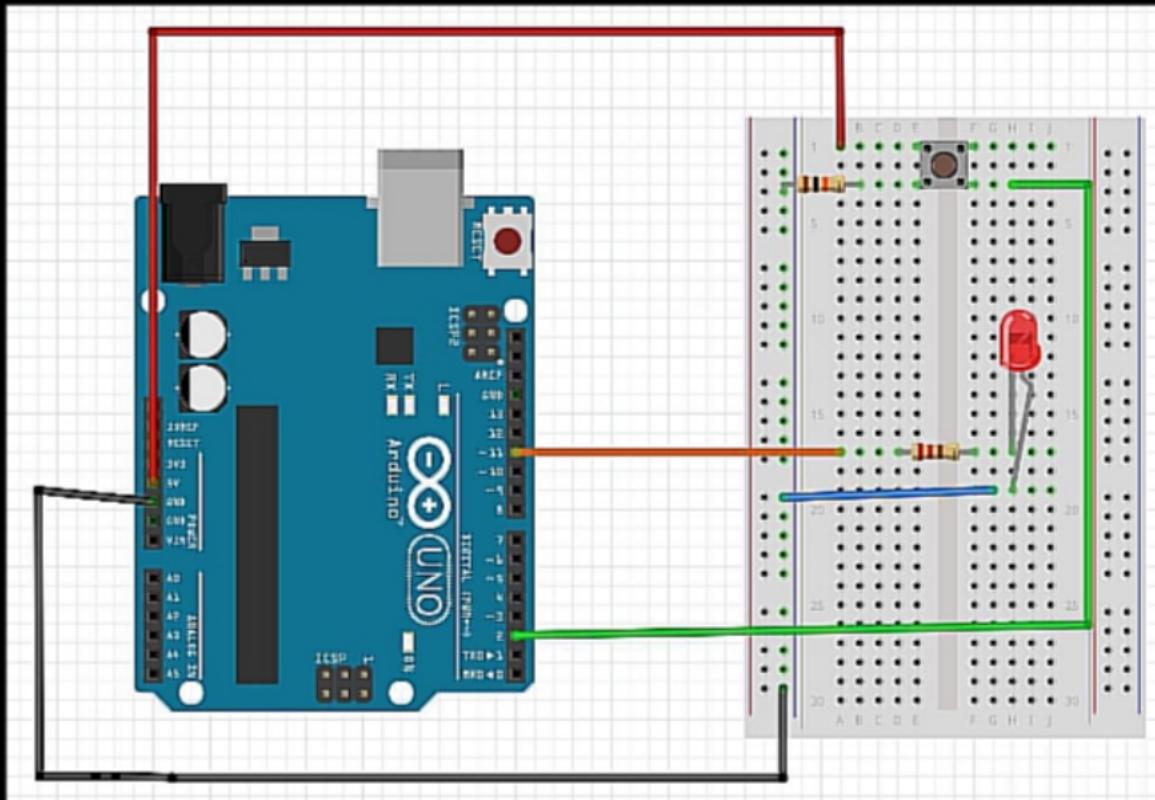
CIRCUITO 4

ACCENSIONE



CIRCUITO 4

SCHEMA FRITZING



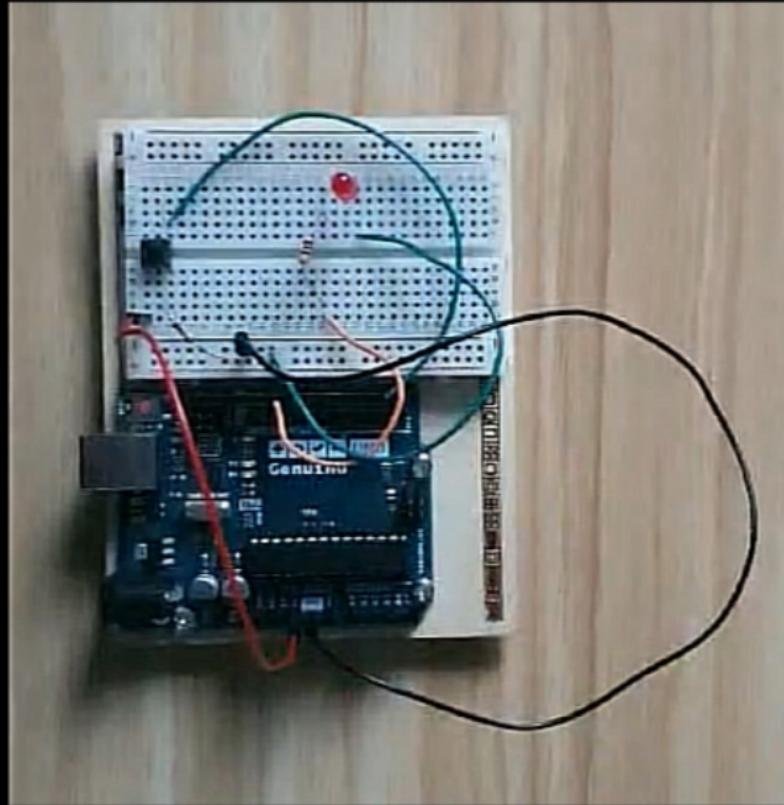
CIRCUITO 4

CODICE

The screenshot shows the Arduino IDE interface with the following details:

- File menu:** File, Edit, Sketch, Tools, Help.
- Sketch menu:** Run, Stop, Refresh, Auto Save On, +.
- Code title bar:** CIRCUITO 4 CODICE
- Code content:** A sketch titled "CIRCUITO 4 CODICE" containing C++ code for a debouncing circuit. The code defines pins, initializes them, and uses a loop to read the state of a button connected to pin 2, debounce it, and then print the state to the serial monitor.
- Output tab:** Shows the output window where the code is being uploaded to the board.
- Bottom status bar:** Line 17, Col 1, Arduino Uno (COM1) (not connected).

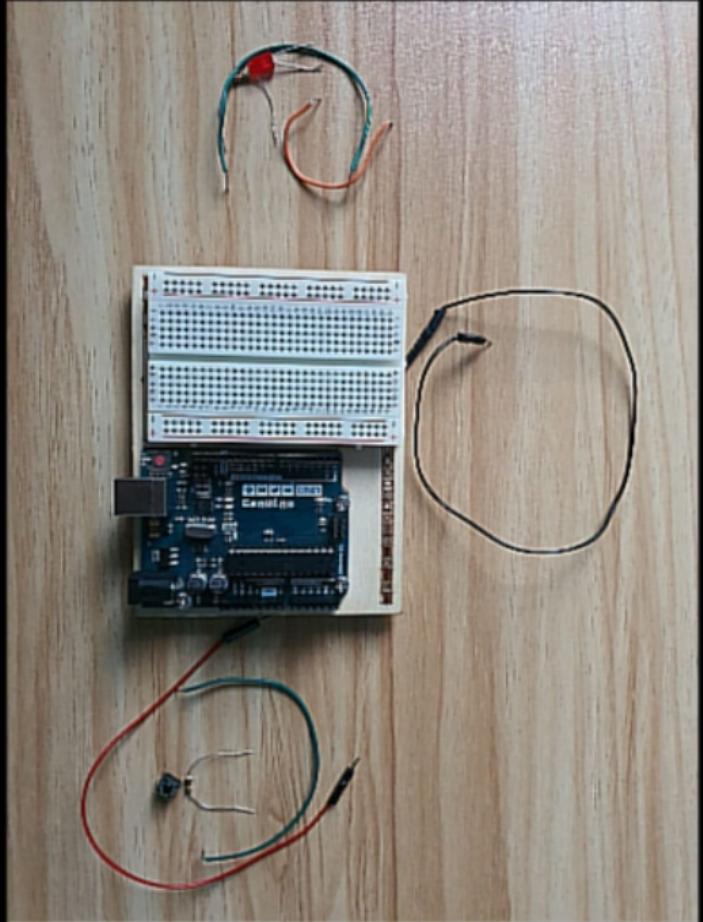
```
const int pinbutton = 2;
const int pinled = 11;
int valbutton=LOW;
int valbuttonOld=LOW;
int ledState = LOW;
long lastdebounceTime = 0;
long debounceDelay = 50;
int reading;
void setup() {
    // put your setup code here, to run once
    pinMode(pinbutton,INPUT);
    pinMode(pinled,OUTPUT);
    Serial.begin(9600);
}
void loop() {
    reading = digitalRead(pinbutton);
    if (reading != valbuttonOld) {
        lastdebounceTime = millis();
    }
    if ((millis() - lastdebounceTime) > debounceDelay) {
        valbutton = reading;
    }
    // put your main code here, to run repeatedly:
    valbutton=digitalRead(pinbutton);
    delay(100);
    if (valbutton==HIGH && valbuttonOld==LOW)
    {
        ledState =! ledState;
    }
    Serial.println(ledState);
    digitalWrite(pinled, ledState);
    valbuttonOld=valbutton;
}
```



CIRCUITO 5

In questo circuito vi spiegheremo come accendere un diodo LED e farlo rimanere acceso con un debouncer, inoltre il monitor seriale dirà "Il LED è spento" se il LED è spento e "Il LED è acceso" se il LED è acceso.

+INFO



CIRCUITO 5

COMPONENTI

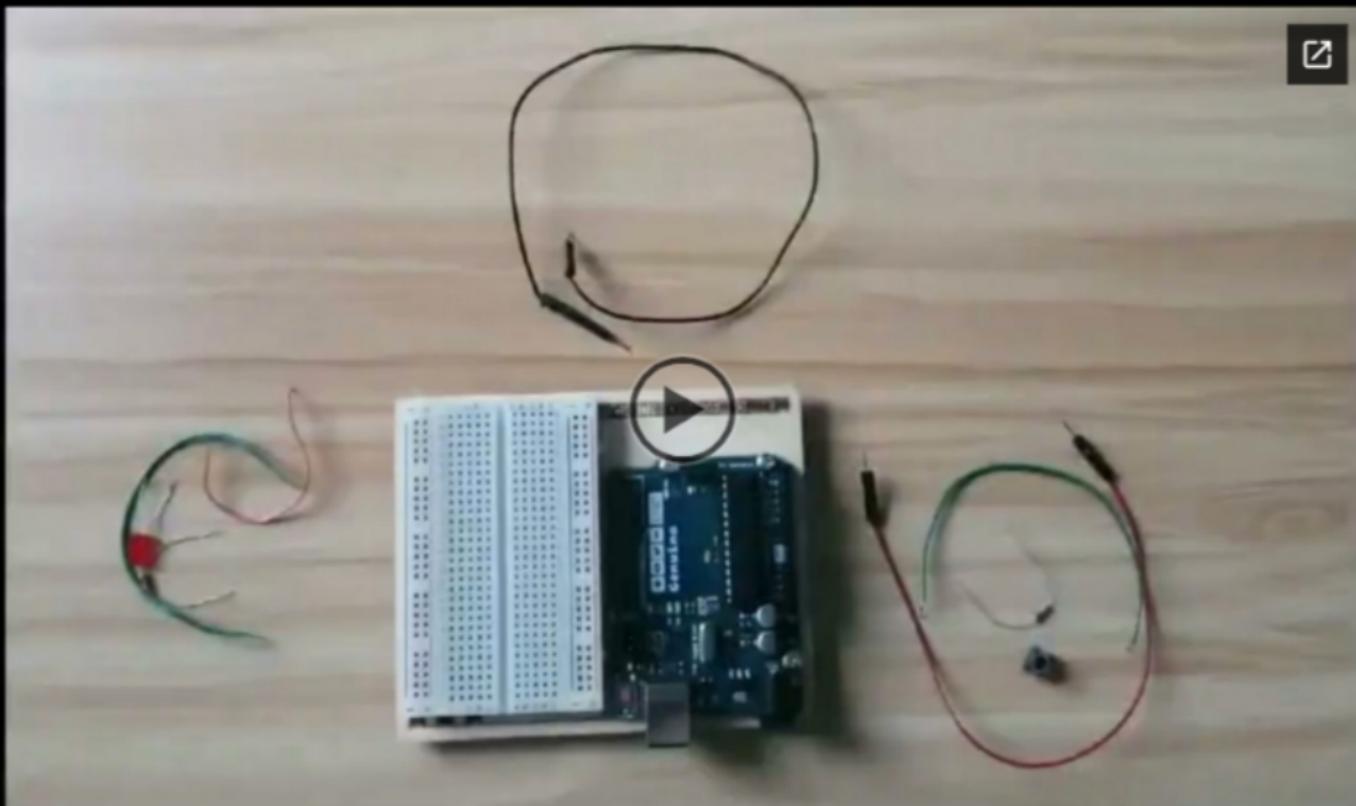
Per realizzare questo circuito,
serviranno:

- Scheda Arduino Uno;
- Breadboard;
- 2x Cavi Jumper;
- Resistenza da $10\text{k}\Omega$;
- Pushbutton;
- Resistenza da 220Ω ;
- 3x Ponticelli

+INFO

CIRCUITO 5

MONTAGGIO



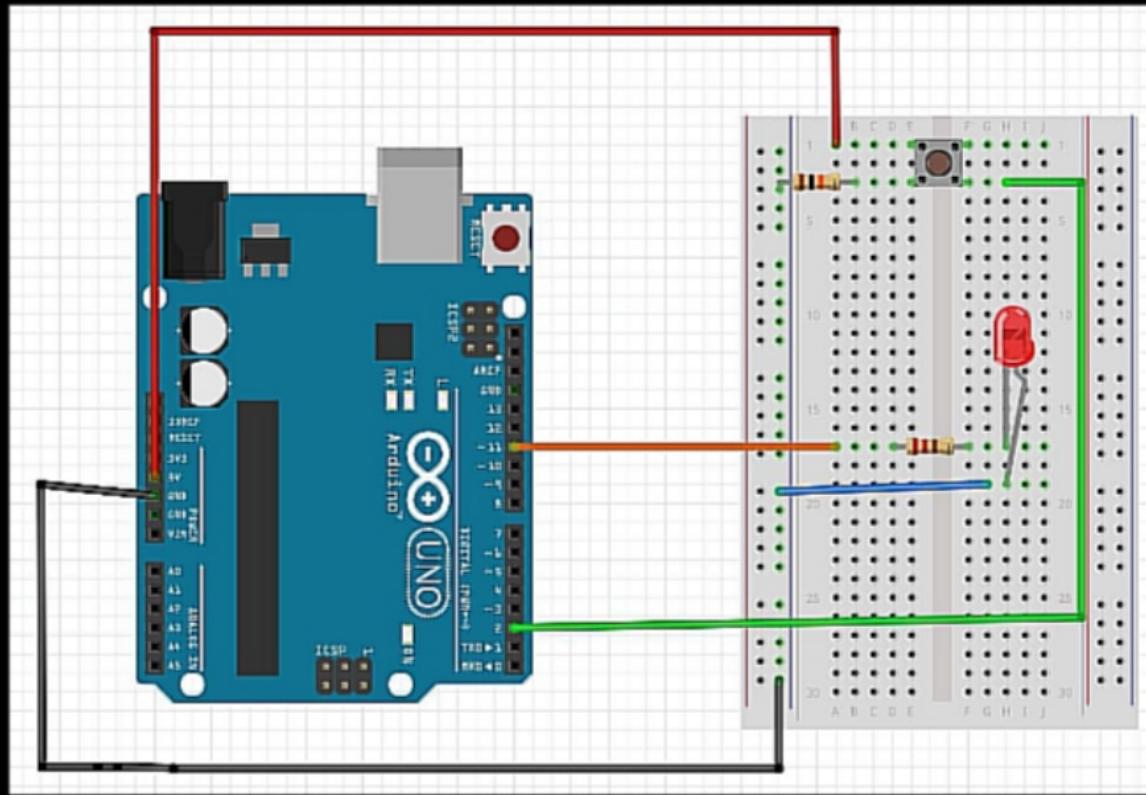
CIRCUITO 5

ACCENSIONE



CIRCUITO 5

SCHEMA FRITZING



CIRCUITO 5

CODICE

File Edit Sketch Tools Help

BOTTONE LED FISSO MONITOR SERIALE ACCESO SPENTO.ino

```
1 const int pindbutton = 2;
2 const int pindled = 11;
3 int valbutton=LOW;
4 int valbuttonold=LOW;
5 int ledstate = LOW;
6
7 void setup() {
8     // put your setup code here, to run once:
9     pinMode(pindbutton,INPUT);
10    pinMode(pindled,OUTPUT);
11    Serial.begin(9600);
12 }
13
14 void loop() {
15     // put your main code here, to run repeatedly:
16     valbutton=digitalRead(pindbutton);
17
18     delay(100);
19     if ((valbutton==HIGH && valbuttonold==LOW)
20     {
21         ledState =! ledState;
22     }
23
24     if(ledState == HIGH)
25     {
26         Serial.println("Il LED è acceso");
27     }
28     else
29     {
30         Serial.println("Il LED è spento");
31     }
32
33     digitalWrite(pindled, ledState);
34     valbuttonold=valbutton;
35 }
36
```

Output

Line 36 Col 1 Arduino Uno or CDM (not connected)

GRAZIE PER L'ATTENZIONE!

