

esempio di codice Arduino per leggere l'input di un bottone collegato al pin 2 dell'Arduino:

```
const int buttonPin = 2; // definisce il pin del bottone
int buttonState = 0;     // variabile per memorizzare lo stato del bottone

void setup() {
  pinMode(buttonPin, INPUT); // imposta il pin del bottone come input
  Serial.begin(9600);        // inizializza la comunicazione seriale
}

void loop() {
  buttonState = digitalRead(buttonPin); // legge lo stato del bottone

  if (buttonState == HIGH) { // se il bottone è premuto
    Serial.println("Bottone premuto!"); // stampa un messaggio sulla porta
    seriale
  }

  delay(50); // ritarda la lettura del bottone per evitare falsi positivi
}
```

In questo esempio, il bottone è collegato al pin 2 dell'Arduino e viene letto tramite la funzione `digitalRead()`. Se il bottone viene premuto, viene stampato un messaggio sulla porta seriale utilizzando la funzione `Serial.println()`. Il ritardo di 50 millisecondi alla fine del loop serve a evitare falsi positivi.

esempio di codice Arduino che include un controllo antirimbalo per il bottone:

```
const int buttonPin = 2; // definisce il pin del bottone
int buttonState = 0;     // variabile per memorizzare lo stato del bottone
int lastButtonState = 0; // variabile per memorizzare lo stato precedente del
bottone
unsigned long lastDebounceTime = 0; // ultimo tempo in cui è stato letto il
bottone
unsigned long debounceDelay = 50;   // tempo minimo tra due letture del
bottone

void setup() {
  pinMode(buttonPin, INPUT); // imposta il pin del bottone come input
  Serial.begin(9600);        // inizializza la comunicazione seriale
}

void loop() {
  int reading = digitalRead(buttonPin); // legge lo stato del bottone

  if (reading != lastButtonState) { // se lo stato del bottone è cambiato
    lastDebounceTime = millis();    // registra l'ultimo tempo in cui è stato
    letto il bottone
  }
}
```

```

    if ((millis() - lastDebounceTime) > debounceDelay) { // se è passato
abbastanza tempo dal controllo precedente
        if (reading != buttonState) { // se lo stato del bottone è ancora
cambiato
            buttonState = reading;          // memorizza il nuovo stato del bottone

            if (buttonState == HIGH) { // se il bottone è premuto
                Serial.println("Bottone premuto!"); // stampa un messaggio sulla
porta seriale
            }
        }
    }

    lastButtonState = reading; // memorizza lo stato precedente del bottone
}

```

In questo esempio, viene utilizzato un algoritmo di controllo antirimbato per eliminare eventuali falsi positivi causati dal rimbalzo meccanico del bottone. Quando viene letto un nuovo valore dal bottone, viene memorizzato il tempo di lettura attuale. Successivamente, se il tempo trascorso dal controllo precedente è maggiore del tempo di debounceDelay, viene confrontato il valore attuale del bottone con l'ultimo stato memorizzato. Se il valore è cambiato, viene aggiornato lo stato del bottone e viene verificato se il bottone è premuto. Solo in questo caso viene stampato un messaggio sulla porta seriale. Infine, viene memorizzato lo stato attuale del bottone come stato precedente per la prossima lettura.

esempio di codice Arduino per controllare un semaforo con un bottone:

```

const int buttonPin = 2; // definisce il pin del bottone
int buttonState = 0;     // variabile per memorizzare lo stato del bottone
int lastButtonState = 0; // variabile per memorizzare lo stato precedente del
bottone
unsigned long lastDebounceTime = 0; // ultimo tempo in cui è stato letto il
bottone
unsigned long debounceDelay = 50;   // tempo minimo tra due letture del
bottone

int currentState = 0; // variabile per memorizzare lo stato attuale del
semaforo

const int redPin = 13; // definisce il pin del LED rosso
const int yellowPin = 12; // definisce il pin del LED giallo
const int greenPin = 11; // definisce il pin del LED verde

void setup() {
    pinMode(buttonPin, INPUT); // imposta il pin del bottone come input
    Serial.begin(9600);        // inizializza la comunicazione seriale
}

```

```

void loop() {
    int reading = digitalRead(buttonPin); // legge lo stato del bottone

    if (reading != lastButtonState) { // se lo stato del bottone è cambiato
        lastDebounceTime = millis(); // registra l'ultimo tempo in cui è stato
        letto il bottone
    }

    if ((millis() - lastDebounceTime) > debounceDelay) { // se è passato
        abbastanza tempo dal controllo precedente
        if (reading != buttonState) { // se lo stato del bottone è ancora
        cambiato
            buttonState = reading; // memorizza il nuovo stato del bottone

            if (buttonState == HIGH) { // se il bottone è premuto

                Serial.println("Bottone premuto!"); // stampa un messaggio sulla
                porta seriale
                Serial.println(currentState);
                currentState++; // passa allo stato successivo
                if (currentState > 3) { // se è l'ultimo stato
                    currentState = 0; // torna allo stato iniziale
                }

            }

        }

    }

}

```

```

lastButtonState = reading; // memorizza lo stato precedente del bottone

```

```

switch (currentState) {
    case 0:
        digitalWrite(redPin, HIGH);
        digitalWrite(yellowPin, LOW);
        digitalWrite(greenPin, LOW);
        break;
    case 1:
        digitalWrite(redPin, HIGH);
        digitalWrite(yellowPin, HIGH);
        digitalWrite(greenPin, LOW);
        break;
    case 2:
        digitalWrite(redPin, LOW);
        digitalWrite(yellowPin, LOW);
        digitalWrite(greenPin, HIGH);
        break;
    case 3:
        digitalWrite(redPin, LOW);

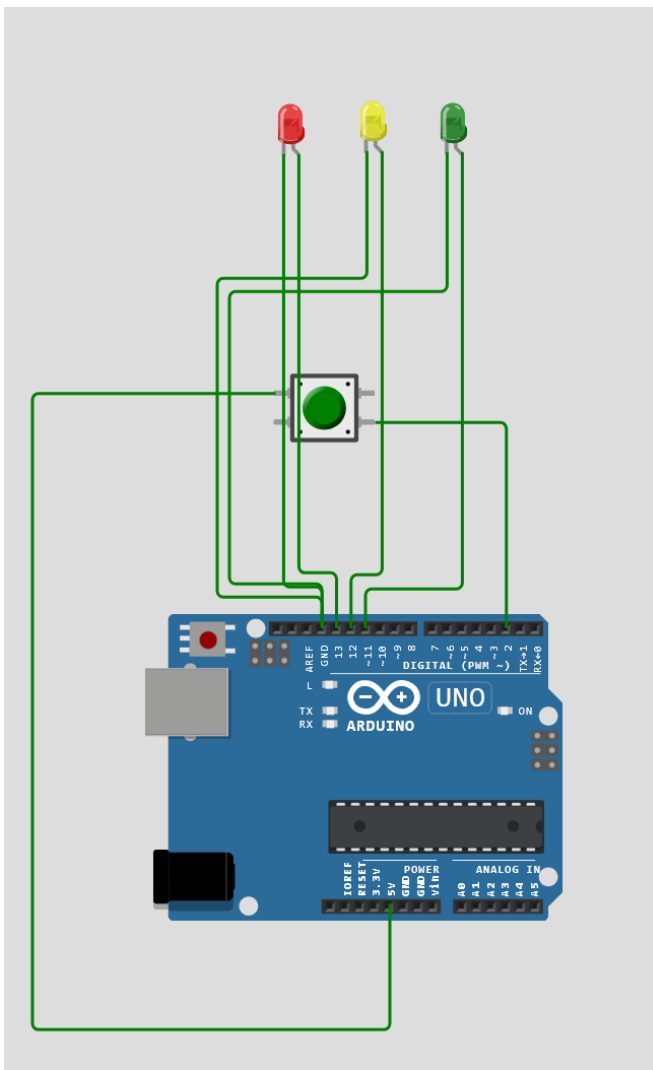
```

```

    digitalWrite(yellowPin, HIGH);
    digitalWrite(greenPin, LOW);
    break;
default:
    break;
}

}

```



Esercizio : fate lampeggiare il giallo