



ARDUINO

ACCENDERE E SPEGNERE LED CON UN PULSANTE

LEZIONE PRECEDENTE

- Abbiamo visto come è facile scrivere e caricare un semplice programma con arduino
- Abbiamo visto come sia facile far lampeggiare il led
- Abbiamo provato a collegare più led

SOLUZIONE

```
1 int pinLedRed = 12;           // il pin che uso è il numero 12
2 int pinLedBlue = 11;        // il pin che uso è il numero 11
3
4 void setup() {
5     pinMode(pinLedRed, OUTPUT); // il pin 12 deve essere un uscita
6     pinMode(pinLedBlue, OUTPUT); // il pin 11 deve essere un uscita
7 }
8
9
10 void loop() {
11     digitalWrite(pinLedRed, HIGH); // Porta in alto l'uscita del pin 12
12     delay(1000); // aspetta 1000ms = 1s
13     digitalWrite(pinLedRed, LOW); // Porta in basso l'uscita del pin 12
14     delay(1000); // aspetta 1000ms = 1s
15
16     digitalWrite(pinLedBlue, HIGH); // Porta in alto l'uscita del pin 12
17     delay(1000); // aspetta 1000ms = 1s
18     digitalWrite(pinLedBlue, LOW); // Porta in basso l'uscita del pin 12
19     delay(1000); // aspetta 1000ms = 1s
20 }
```

ACCENDIAMO UN LED CON UN INTERRUTTORE

Io voglio accendere un led premendo un pulsante

Avrò bisogno di:

1. Poter accendere un led
2. Poter leggere il valore on | off di un interruttore
3. Se l'interruttore è ON allora accendi il led se l'interruttore è OFF allora spegni il led => serve un logica... iniziamo da questa !



LA PRIMA LOGICA

ISTRUZIONE CONDIZIONALE «IF AND ELSE»

PROF. NACLERIO PASQUALE

VALORI BOOLEANI

- Nella logica esistono solo due tipi di possibili valori

VERO e FALSO

- In programmazione abbiamo TRUE | FALSE o 1 | 0 posso usarli entrambi
- Questo tipo di valori ha una variabile tutta sua si chiama BOOLEAN

```
bool nome_variabile = valore_variabile;
```

```
bool x = 1; //la variabile x è vera
```

```
bool y = false; //la variabile y è falsa
```

```
bool z = HIGH; //la variabile è alta e cioè 1
```

HIGH E LOW

Con Arduino possiamo usare le parole HIGH e LOW
al posto di TRUE o FOLSE oppure di 1 o 0

VOGLIAMO FAR ACCADERE QUALCOSA SOLO IN UN CASO SPECIFICO

- Questo tipo di istruzioni si chiama ISTRUZIONI CONDIZIONALI
- Voglio eseguire del codice solo se si verifica una certa condizione

```
Se si verifica la ( condizione ) {
```

```
    // allora viene eseguito questo blocco di istruzioni 1 (notare le graffe)
```

```
    } altrimenti {
```

```
        // viene eseguito questo blocco di istruzioni 2 (notare le graffe)
```

```
    }
```


SINTASSI

- Si userà la parola inglese «if» che significa «se»
- Si userà la parola inglese «else» che significa «altrimenti»
- **ATTENZIONE «ELSE» NON E' OBBLIGATORIA!**
- **ATTENZIONE ALLE PARENTESI**

```
1  if (condizione) {  
2  
3    // blocco di istruzioni 1  
4  
5  } else {  
6  
7    // blocco di istruzioni 2  
8  
9  }
```

```
1 if (condizione) {
2
3 // blocco di istruzioni 1
4
5 } else {
6
7 // blocco di istruzioni 2
8
9 }
```

Se si verifica la (condizione) {

// allora viene eseguito questo blocco di istruzioni 1 (notare le graffe)

} **altrimenti** {

// viene eseguito questo blocco di istruzioni 2 (notare le graffe)

}

LE CONDIZIONI

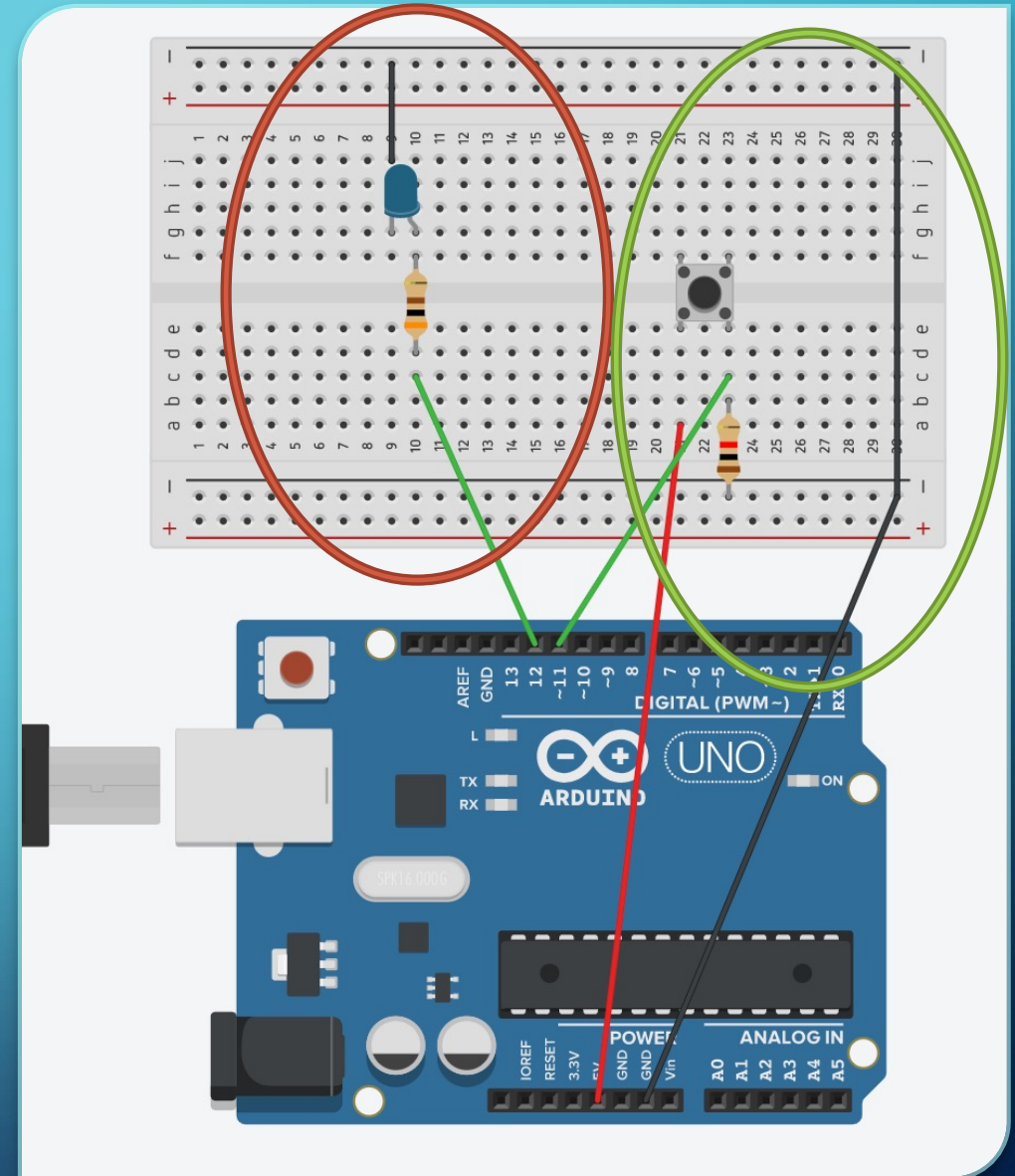
== uguaglianza	Se $x = 3$, allora	$x == 4$ è falso
!= diversità	Se $x = 3$, allora	$x != 4$ è vero
> maggiore di	Se $x = 3$, allora	$x > 4$ è falso
< minore di	Se $x = 3$, allora	$x < 4$ è vero
>= maggiore o uguale	Se $x = 3$, allora	$x >= 4$ è falso
<= minore o uguale	Se $x = 3$, allora	$x <= 4$ è vero

IL CIRCUITO

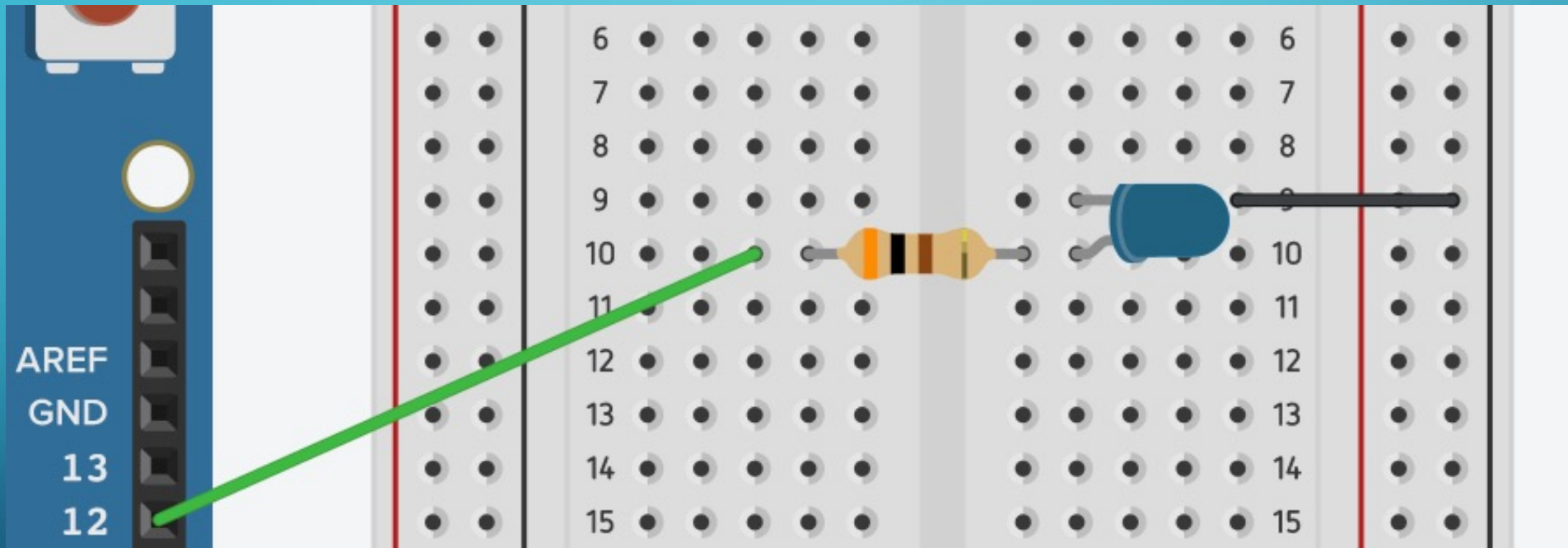
- ATTENZIONE A COME COSTRUIAMO IL CIRCUITO
- Ci sono 2 circuiti distinti

- Ora vedremo in dettaglio ogni parte

PROF. NACLERIO PASQUALE



IL SOLITO LED

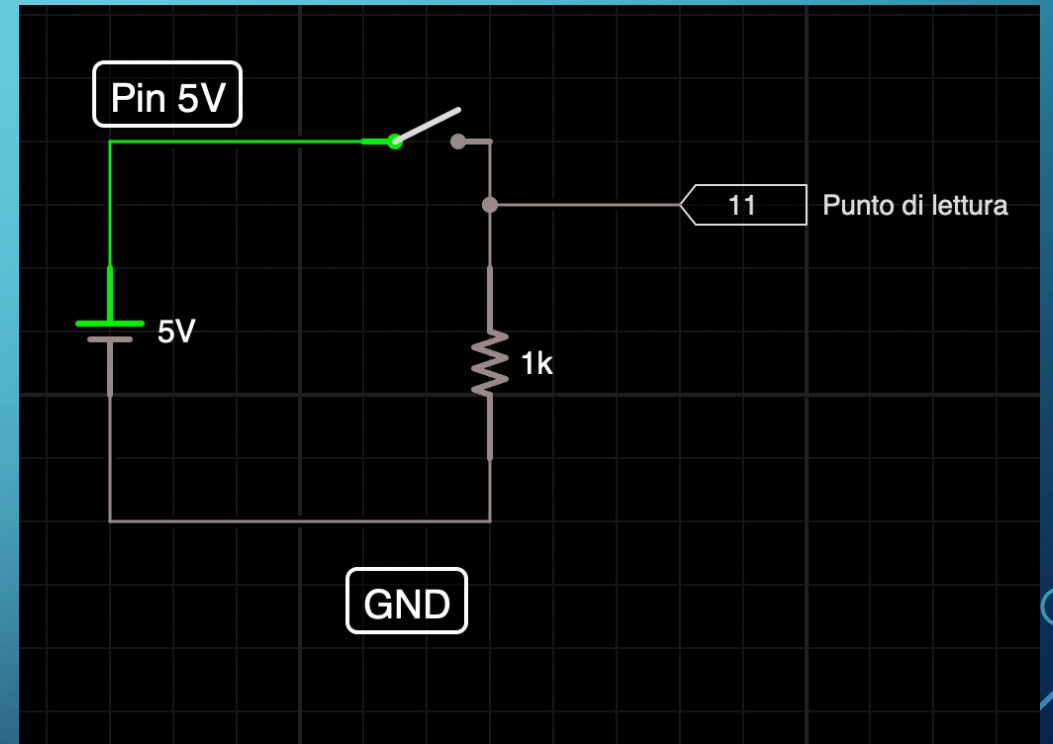
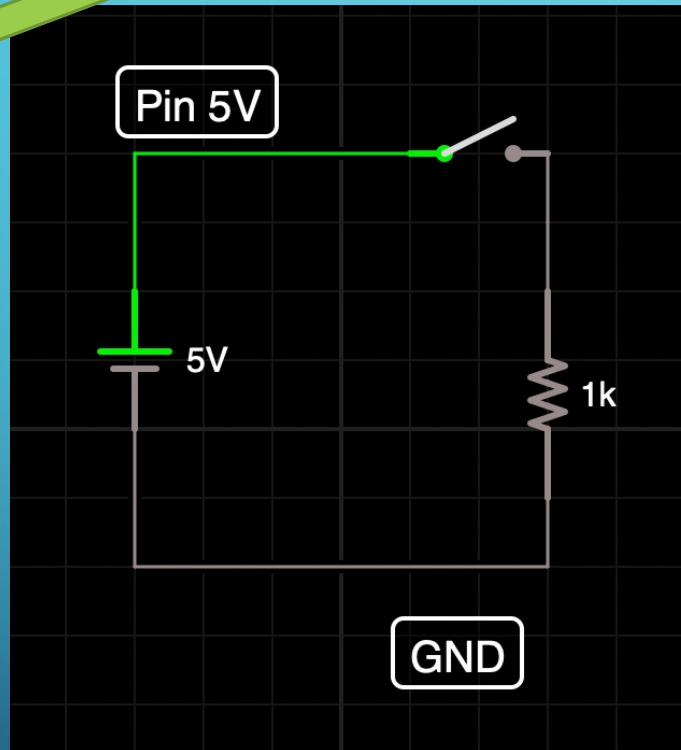
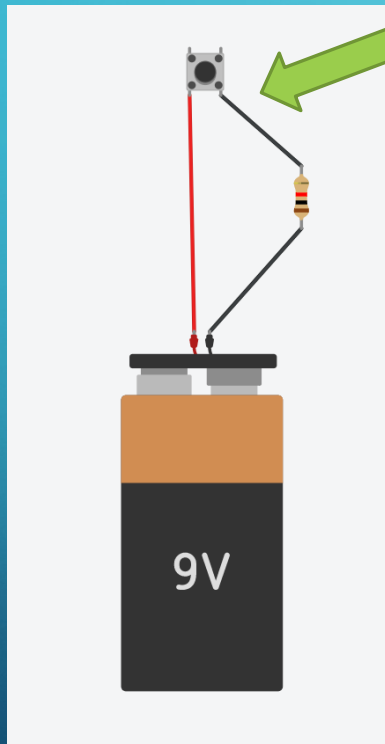


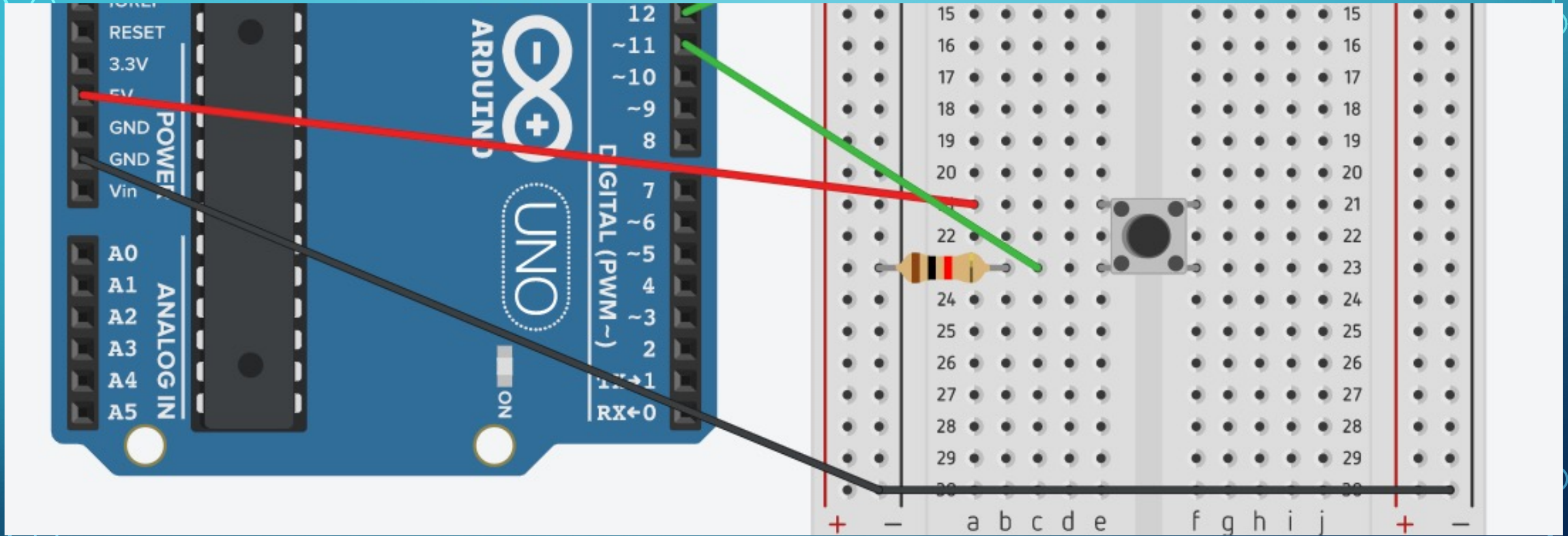
GND

Pin 12

INTERRUTTORE

Qui dovremo mettere la lettura del pin 11





ISTRUZIONI DI LETTURA

- L'istruzione di lettura da un pin digitale è:

```
digitalRead(numero_del_pin)
```

- Come si usa ?

Dato che noi leggiamo un valore dobbiamo metterlo da qualche parte e quindi in una variabile...

```
bool lettura = digitalRead(numero_del_pin);
```


SETUP

- Dentro al setup dovremo impostare il pin scelto come INPUT perché riceveremo un segnale

```
6 void setup() {  
7   pinMode(pinLed, OUTPUT); //settiamo il pin del led a ingresso  
8   pinMode(pinSwitch, INPUT); //settiamo il pin dell'interruttore a uscita  
9 }  
10
```

- Noi usiamo anche il pin per il led che invece deve essere un uscita e quindi va messo ad OUTPUT

SCRIVIAMOLO A PAROLE

Io vorrei che il mio Arduino facesse queste operazioni:

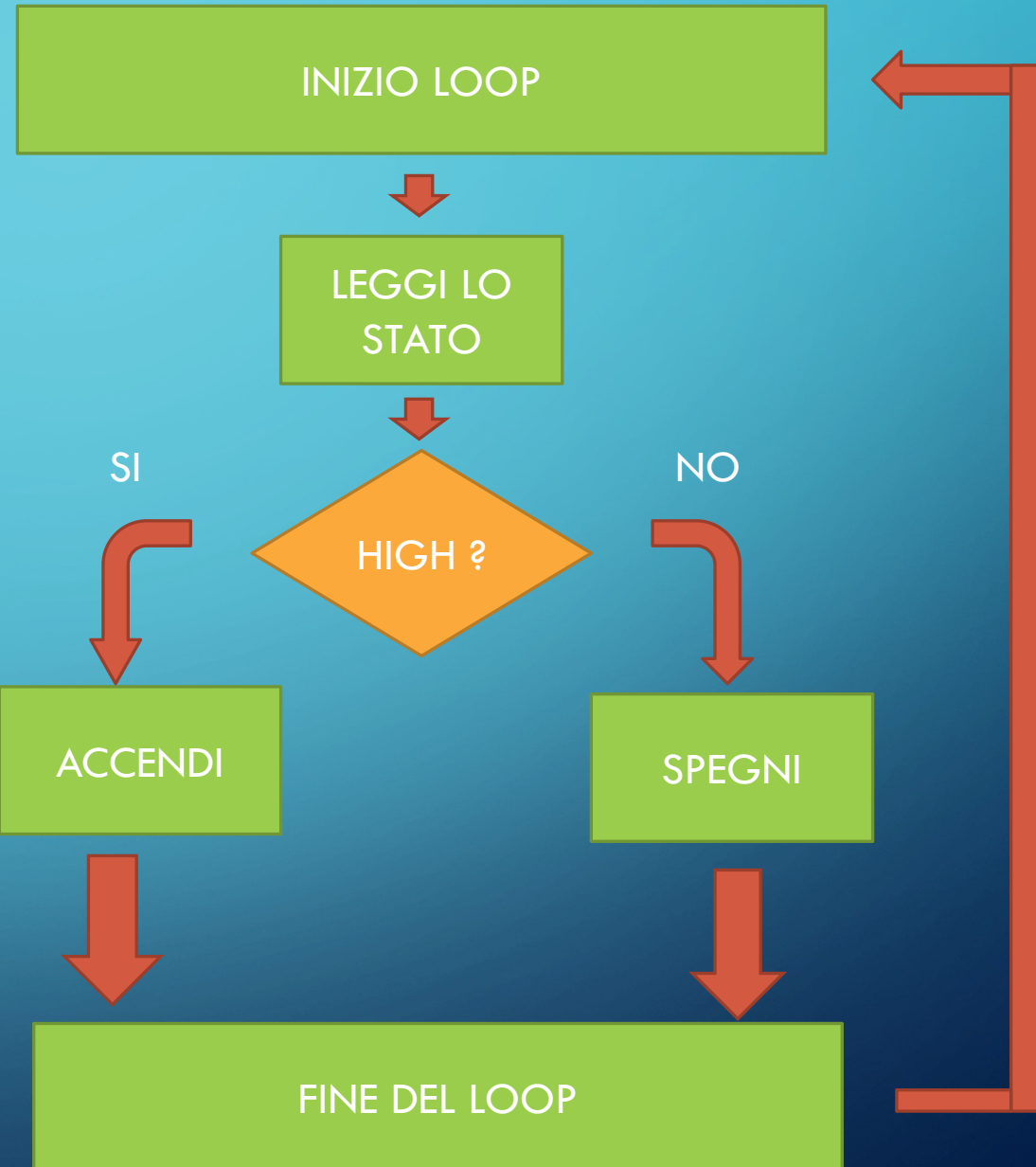
1. Leggi l'interruttore e controlla se è ad un livello alto o basso
2. Se il livello è alto accendi il led
3. Altrimenti spegni il led

ECCO IL CODICE

```
8
9 void loop() {
10   bool lettura = digitalRead(pinSwitch); //leggiamo il valore del pin 11 e salviamolo
11
12   if(lettura == HIGH){                //se la lettura è alta
13
14     digitalWrite(pinLed, HIGH);      //accendi il led
15     delay(100);                       //aspetta 100 ms il tempo di rilasciare il dito
16
17   }
18   else{                                //altrimenti
19
20     digitalWrite(pinLed, LOW);       //tieni spento il led
21
22   }
23
24 }
```

ANALIZZIAMOLO A BLOCCHI

```
9 void loop() {  
10   bool lettura = digitalRead(pinSwitch);  
11  
12   if(lettura == HIGH){  
13  
14     digitalWrite(pinLed, HIGH);  
15     delay(100);  
16  
17   }  
18   else{  
19  
20     digitalWrite(pinLed, LOW);  
21  
22   }  
23 }  
24 }
```



The background is a dark blue gradient. In the corners, there are white line-art graphics resembling circuit traces or neural network connections, with small circles at the end of the lines.

ACCADE PER COSÌ POCO TEMPO CHE RIMANE
SPENTO QUASI SEMPRE

PROF. NACLERIO PASQUALE

ECCO IL CODICE COMPLETO

```
Interruttore
1 int pinLed = 12;    //pin per il led
2 int pinSwitch = 11; //pin per l'interruttore
3
4 void setup() {
5   pinMode(pinLed, OUTPUT); //settiamo il pin del led a ingresso
6   pinMode(pinSwitch, INPUT); //settiamo il pin dell'interruttore a uscita
7 }
8
9 void loop() {
10  bool lettura = digitalRead(pinSwitch); //leggiamo il valore del pin 11 e salviamolo
11
12  if(lettura == HIGH){                //se la lettura è alta
13
14     digitalWrite(pinLed, HIGH);      //accendi il led
15     delay(100);                      //aspetta 100 ms il tempo di rilasciare il dito
16
17  }
18  else{                                //altrimenti
19
20     digitalWrite(pinLed, LOW);       //tieni spento il led
21
22  }
23
24 }
```



VOGLIAMO DI PIÙ
VOGLIO UN VERO INTERRUTTORE STABILE

PROF. NACLERIO PASQUALE

COSA VOGLIO

- Ad ora il mio led si accende quando premo, ma rimane spento
- lo voglio un sistema che quando l'interruttore viene premuto il led si accende e si spegne solo quando lo premo nuovamente

PROVIAMO A SCRIVERE A PAROLE LA LOGICA

Creiamo uno stato che è LOW

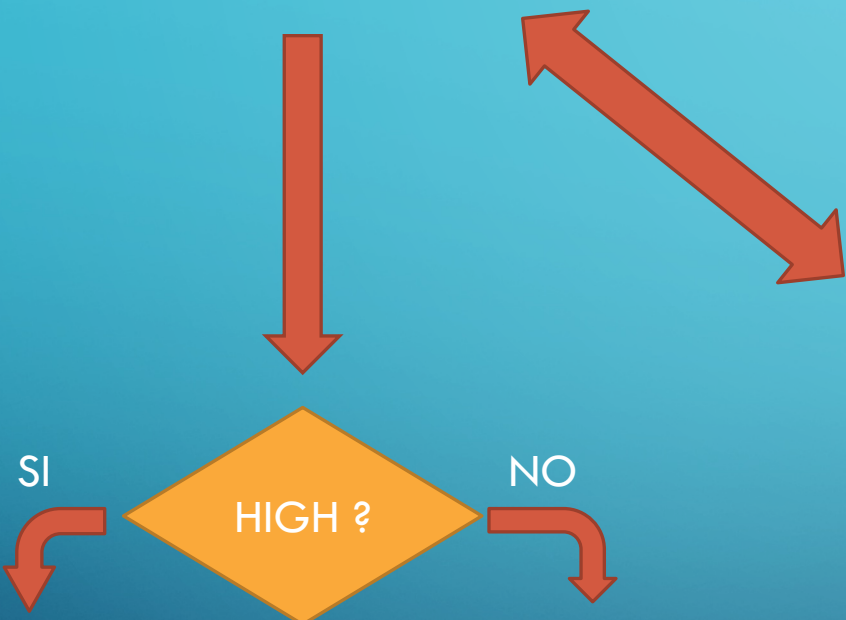
Se premo il pulsante allora cambio lo stato

Se lo stato è HIGH allora tieni acceso il led

Se lo stato è LOW allora spegni il led

VARIABILE DI CONTROLLO

Stato



QUESTO è UN ALTRO IF

LEGGI LO STATO



Stato = HIGH se era prima LOW
Stato = LOW se era prima HIGH

Lascia tutto come era prima

QUESTO è UN IF

CREO LE VARIABILI

Interruttore_con_stato §

```
1 int pinLed = 12;           //pin per il led
2 int pinSwitch = 11;       //pin per l'interruttore
3 bool lettura;             //creo un contenitore dove mettere la lettura del pulsante
4 bool stato = LOW;         //imposto uno stato a basso;
5
6 void setup() {
```

LA LOGICA

```
11 void loop() {
12   lettura = digitalRead(pinSwitch); //leggiamo il valore del pin 11 e salviamolo
13
14   if(lettura == HIGH){              //se la lettura è alta
15     stato = !stato;                 //cambia lo stato
16     delay(200);                     //aspetta qualche secondo
17   }
18
19   if(stato == HIGH){                //se lo stato è alta
20
21     digitalWrite(pinLed, HIGH);     //accendi il led
22
23   }
24   else{                              //altrimenti
25
26     digitalWrite(pinLed, LOW);      //tieni spento il led
27
28   }
29
30 }
```

```
Interruttore_con_stato §
1 int pinLed = 12; //pin per il led
2 int pinSwitch = 11; //pin per l'interruttore
3 bool lettura; //creo un contenitore dove mettere la lettura del pulsante
4 bool stato = LOW; //imposto uno stato a basso;
5
6 void setup() {
7   pinMode(pinLed, OUTPUT); //settiamo il pin del led a ingresso
8   pinMode(pinSwitch, INPUT); //settiamo il pin dell'interruttore a uscita
9 }
10
11 void loop() {
12   lettura = digitalRead(pinSwitch); //leggiamo il valore del pin 11 e salviamolo
13
14   if(lettura == HIGH){ //se la lettura è alta
15     stato = !stato; //cambia lo stato
16     delay(200); //aspetta qualche secondo
17   }
18
19   if(stato == HIGH){ //se lo stato è alta
20
21     digitalWrite(pinLed, HIGH); //accendi il led
22
23   }
24   else{ //altrimenti
25
26     digitalWrite(pinLed, LOW); //tieni spento il led
27
28   }
29 }
30 }
```