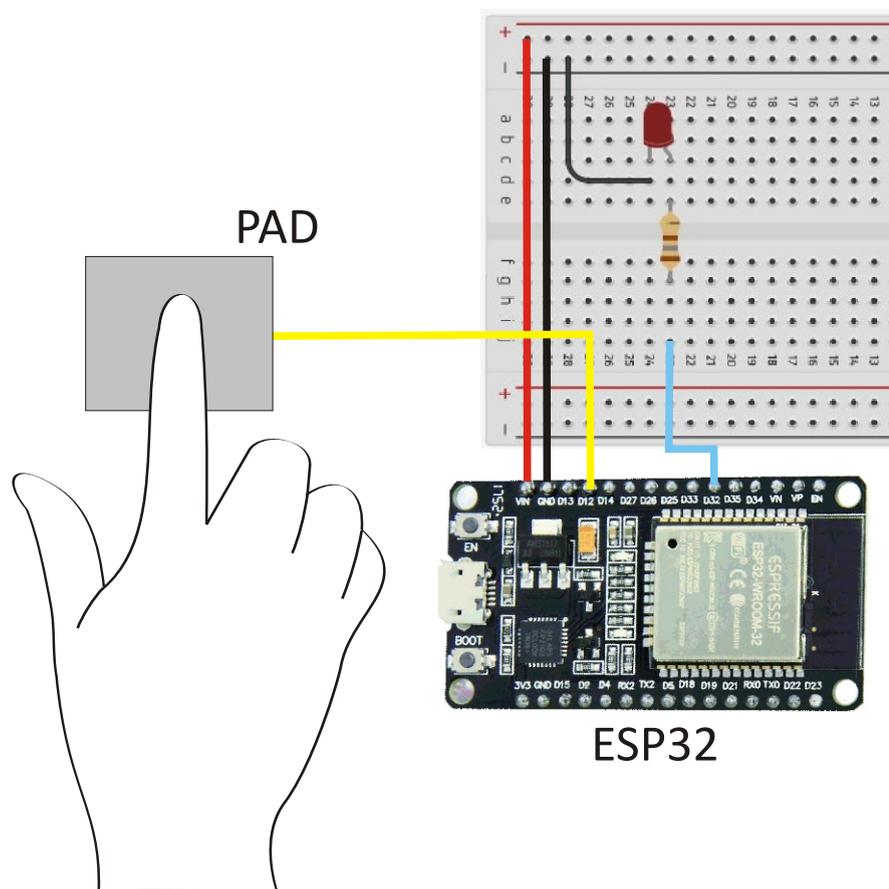


# Gli approfondimenti di

# @BOROBOTICA

## ESP 32 Sensori capacitivi

Ipotizziamo di dover lavorare in ambienti ostili, con presenza di polveri o di grasso, come in un'officina. In casi come questi, utilizzare dei pulsanti meccanici può essere controproducente, visto che potrebbero deteriorarsi in breve tempo. Una soluzione potrebbe essere il pad capacitivo. E' questo una placchetta di metallo che se viene toccata fa in modo di avviare un processo (accensione led, scatto di un relè, etc.). Il nostro ESP32 ha ben 9 pin, che debitamente configurati possono fare al caso nostro. Consultando il pinout della scheda possiamo vedere che i piedini sono i GPIO 2, 4, 12, 13, 14, 15, 27, 32, 33. Sceglieremo il 12.



Ed ecco lo schema di collegamento. Il pad può essere un pezzo di lamiera o un qualsiasi materiale conduttivo. Il circuito funziona bene anche con un filo spellato in punta, o addirittura un jumper utilizzato come sensore. La scheda (visto che i collegamenti sono tutti da un lato) si può inserire direttamente nel breadboard. L'alimentazione si può prendere direttamente dal cavo USB.

Ecco lo sketch da caricare sull'ESP32. Ricordiamoci di premere, durante il caricamento, il pulsante "BOOT".

```
CAP_SENS
1 //pad capacitivi
2 const int touchPin = 12;//collegato al Touch5
3 const int ledPin = 32;
4 const int soglia = 50;// stabilisce il valore di soglia
5 void setup()
6 {
7   pinMode(ledPin, OUTPUT); //il pin del led in uscita
8   Serial.begin(115200);
9   delay(3000);
10
11 }
12
13 void loop()
14 {
15   int valoreCapacitivo =touchRead(touchPin);
16   //attribuisce alla variabile valoreCapacitivo il valore letto sul pin
17   Serial.print("valore:");
18   Serial.println(valoreCapacitivo); //scrive sulla seriale il valore letto
19   if(valoreCapacitivo <= soglia)//fa il confronto con la variabile "soglia"
20   {
21     Serial.print("toccato:");
22     digitalWrite(ledPin, HIGH);// led acceso
23   }
24   else
25   {
26     digitalWrite(ledPin, LOW);// led spento
27   }
28
29 }
```

Lo sketch definisce le variabili, poi imposta in uscita il pin relativo al led (32).

Nel void loop assegna una variabile locale "valoreCapacitivo" che viene letta tramite il comando "touchRead" sul pin dedicato (12). Questo valore viene poi confrontato con quello di soglia.