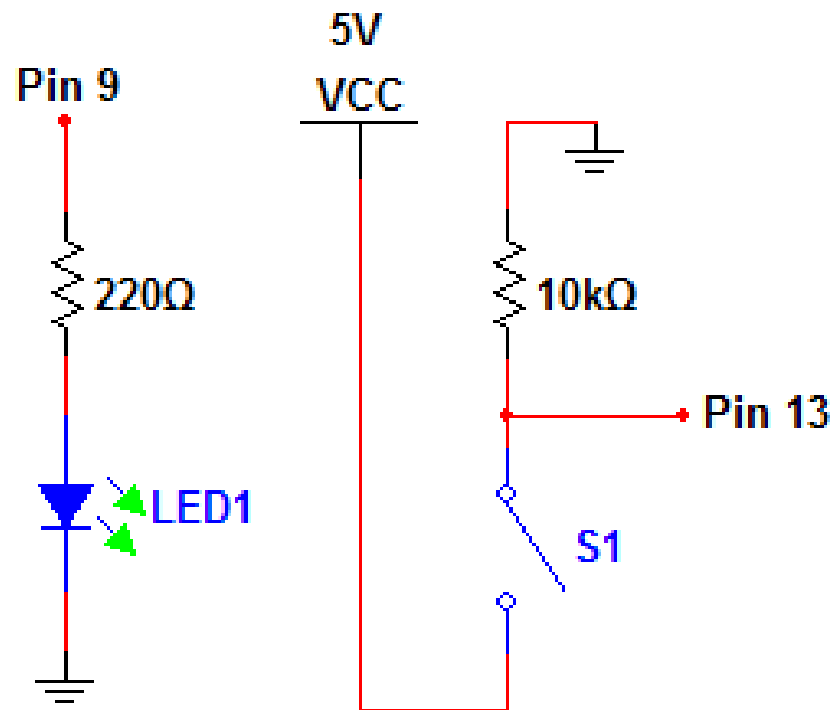


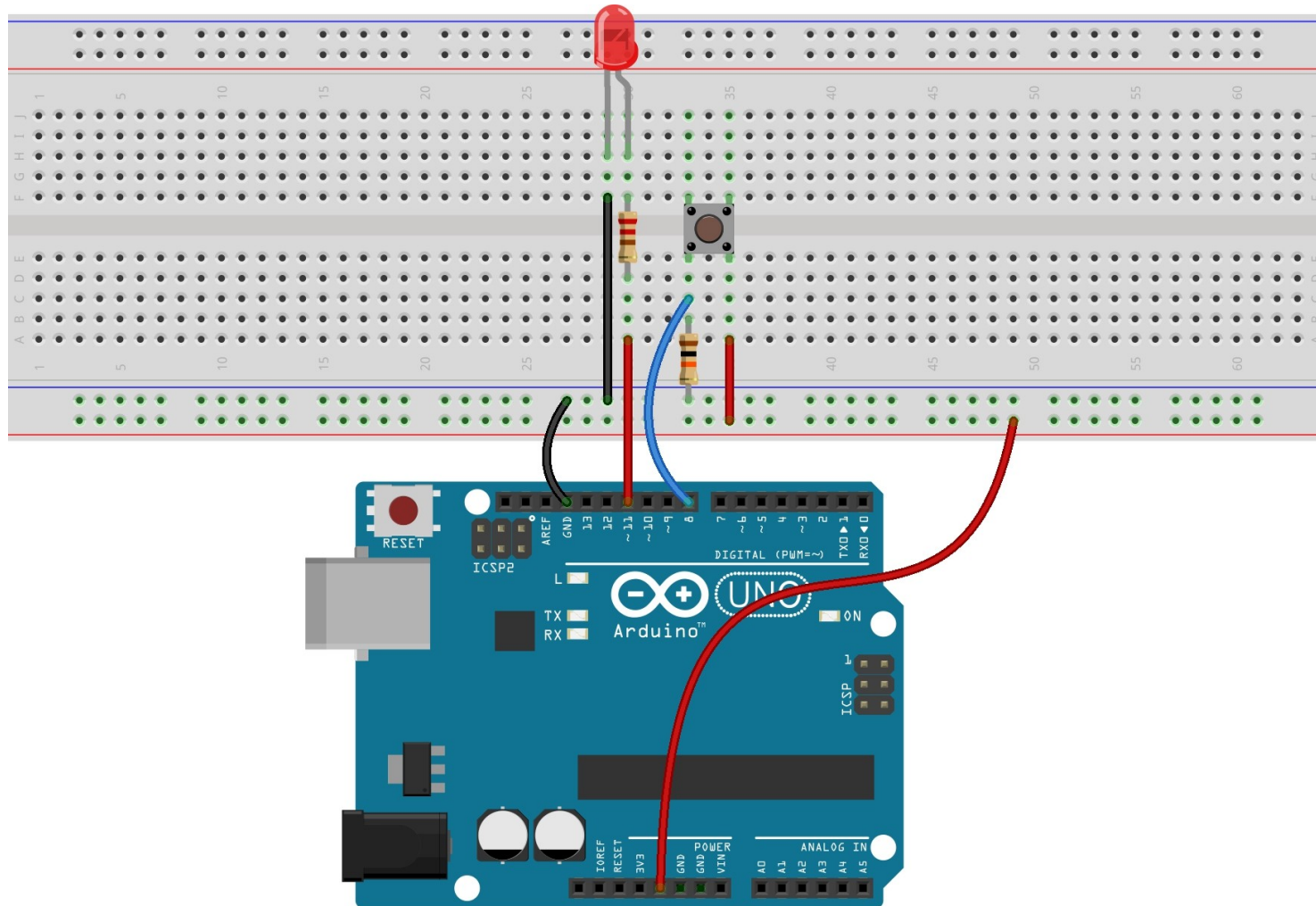
Pulsante e LED

Schema



Pulsante e LED

Circuito



Riconoscere la pressione

- Con il monitor seriale vediamo le istruzioni nella funzione *loop()*
 - ogni riga con l'output **on** oppure **off**
- Ogni ciclo lo sketch verifica se il pulsante è premuto o meno e lo comunica
- È più utile identificare solo il cambiamento di stato del pulsante
 - istante in cui viene premuto: **evento**

Riconoscere la pressione

- Il passaggio di un segnale elettrico
 - da LOW a HIGH è definito rising edge;
 - da HIGH a LOW è chiamato falling edge.
- Lo sketch può distinguere questi due eventi
 - corrispondono a pressione e rilascio del pulsante
- attribuire un comportamento specifico
- ignorare uno dei due se necessario.

Riconoscere la pressione

- Una variabile per memorizzare lo stato del pulsante nel ciclo precedente
- Confrontarlo con lo stato attuale per verificare se è cambiato.
- Dichiariamo all'inizio dello sketch:

int prevButtonState = LOW;

// memorizza lo stato precedente del pulsante

Riconoscere la pressione

- Aggiungere una condizione *if*
- Se lo stato del pin è cambiato rispetto al ciclo precedente:

// condizione if per riconoscere pressione e rilascio:

if (buttonState != prevButtonState) {

- Il codice contenuto tra le parentesi graffe viene eseguito solo quando il pulsante cambia stato:
 - da LOW a HIGH o viceversa

Riconoscere la pressione

- Aggiungere una istruzione per aggiornare il valore della variabile ***prevButtonState*** attribuendole il valore ***buttonState*** corrente
 - In questo modo sarà disponibile per eseguire il confronto nel ciclo successivo:

prevButtonState = ***buttonState***;

- Dal monitor seriale appariranno i messaggi on e off solo alla pressione e al rilascio del pulsante.

Comportamento differente

- Tramite i pin possiamo collegare sensori e attuatori
- Con le istruzioni decidiamo come interpretare gli input e quali output eseguire
- Lo stesso circuito può avere comportamenti diversi in base alla logica definita nello sketch trasferito sulla scheda
- Senza modificare il circuito possiamo fare in modo che il LED si illumini a una prima pressione e resti acceso spegnendosi alla pressione successiva

Comportamento differente

- Distinguere la pressione (rising edge) e il rilascio (falling edge) del pulsante
- Per fare in modo che il LED resti acceso al rilascio del pulsante reagiremo al solo rising edge
- Concatenare due verifiche in un'unica condizione if
 - *If (buttonState != prevButtonState && buttonState == HIGH) {*
- Tra le due condizioni il simbolo **&&** bequivale a un and logico
 - Le istruzioni contenute nel blocco if saranno eseguite solo se lo stato del pulsante è cambiato (prima condizione) e il nuovo stato del pulsante è HIGH (seconda condizione)

GLI OPERATORI BOOLEANI

Gli equivalenti di **e** (*and*), **o** (*or*) e **non** (*not*)

- **&&** (**and**): restituisce *true* se tutte le condizioni sono valide;
- **||** (**or**): restituisce *true* se almeno una delle condizioni è valida;
- **!** (**not**): restituisce true se la condizione seguente non è valida.

Comportamento differente

// dichiarazione delle variabili:

int buttonState = 0;

// memorizza lo stato del pulsante

int prevButtonState = LOW;

// memorizza lo stato precedente del pulsante

int ledState= LOW;

// memorizza lo stato del LED

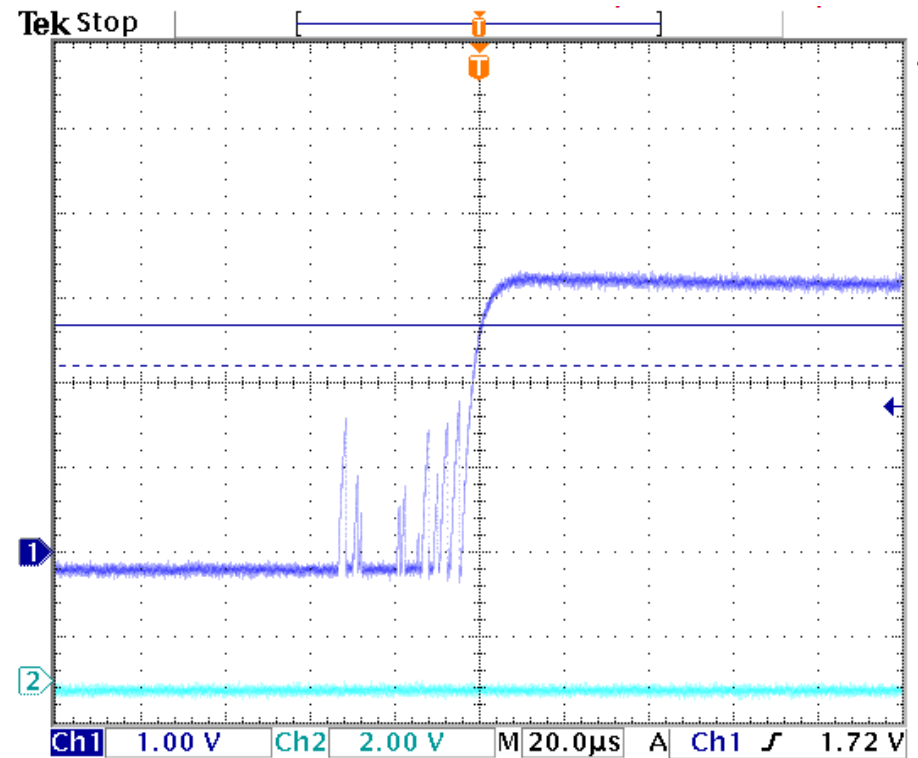
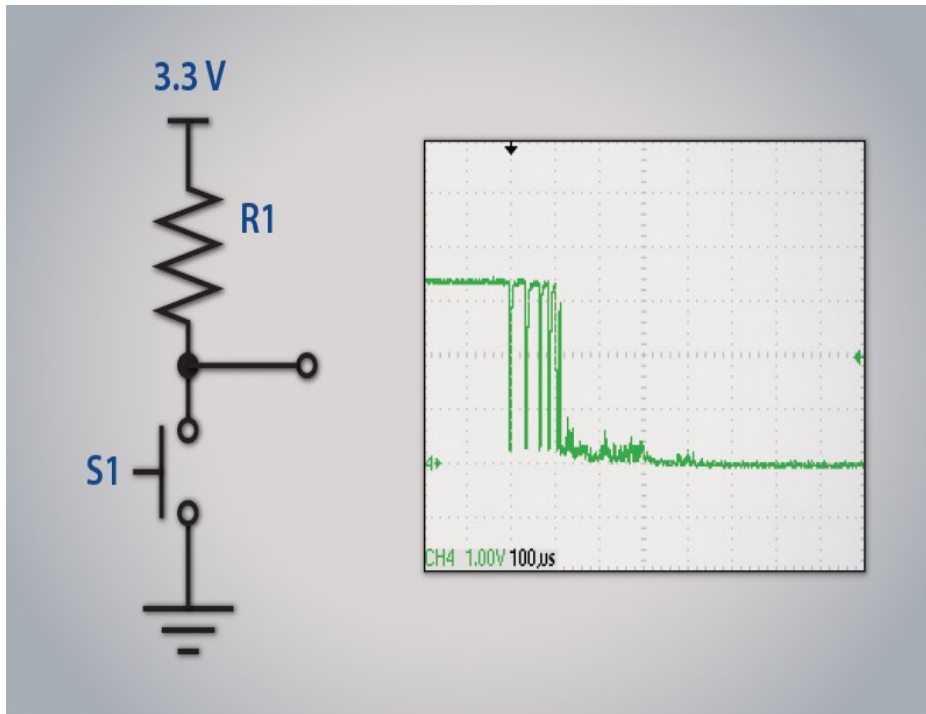
Comportamento differente

```
void loop() {  
    buttonState = digitalRead(buttonPin); // lettura del pin a cui è connesso il pulsante:  
    if (buttonState != prevButtonState && buttonState == HIGH) { // condizione combinata:  
        ledState = !ledState; // inversione dello stato del LED  
        If (ledState == HIGH) {  
            digitalWrite(ledPin, HIGH);  
            Serial.println("on");  
        }  
        else {  
            digitalWrite(ledPin, LOW);  
            Serial.println("off");  
        }  
    }  
    prevButtonState = buttonState;  
}
```

Debounce del pulsante

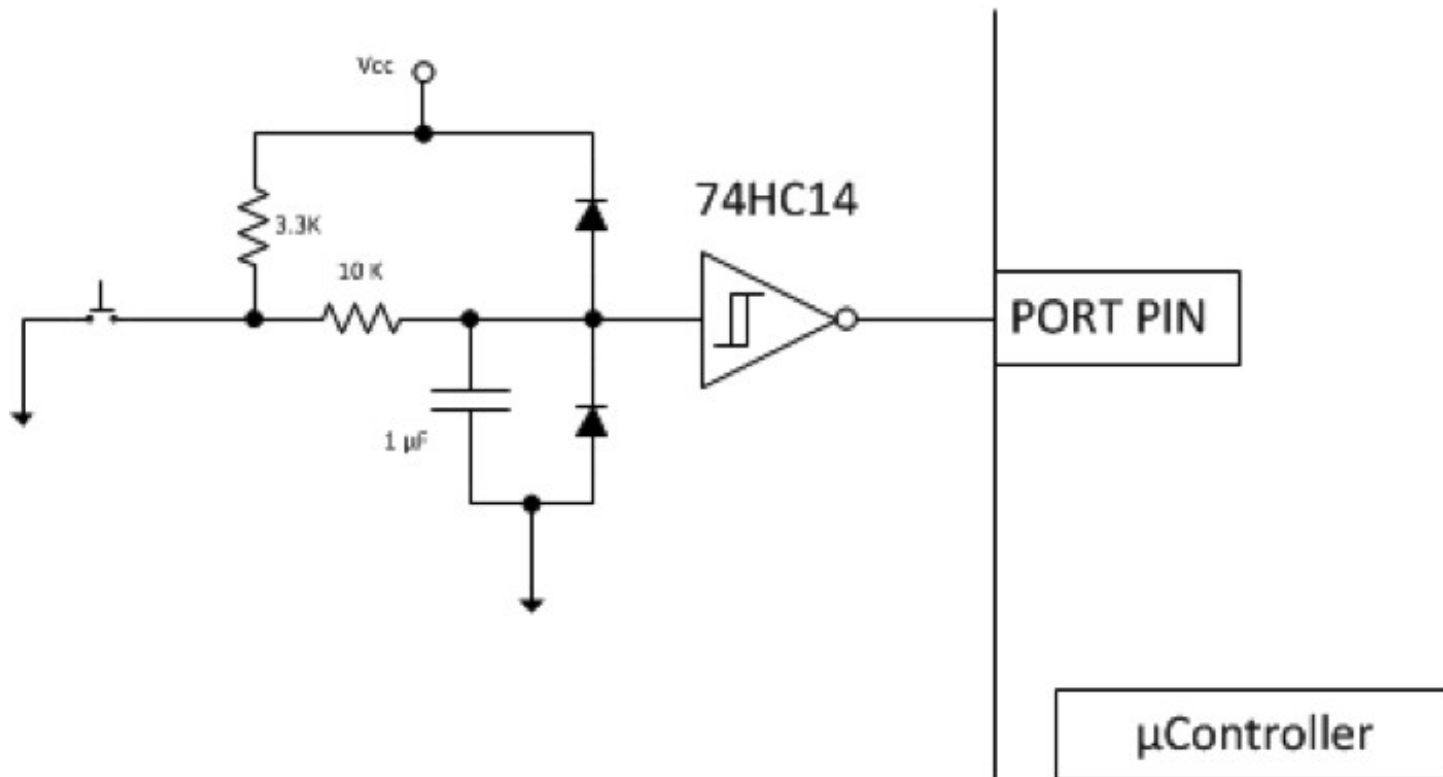
- A volte il comportamento non è quello che ci aspetteremmo:
 - Il LED si accende o si spegne anche quando non dovrebbe
- Il pulsante al suo interno ha una sottile piastra metallica che scatta in posizione di contatto quando esercitiamo pressione e torna elasticamente nella posizione di riposo
- Questa elasticità crea falsi contatti
 - In inglese questo comportamento è chiamato bounce (rimbalzo)

Debounce del pulsante



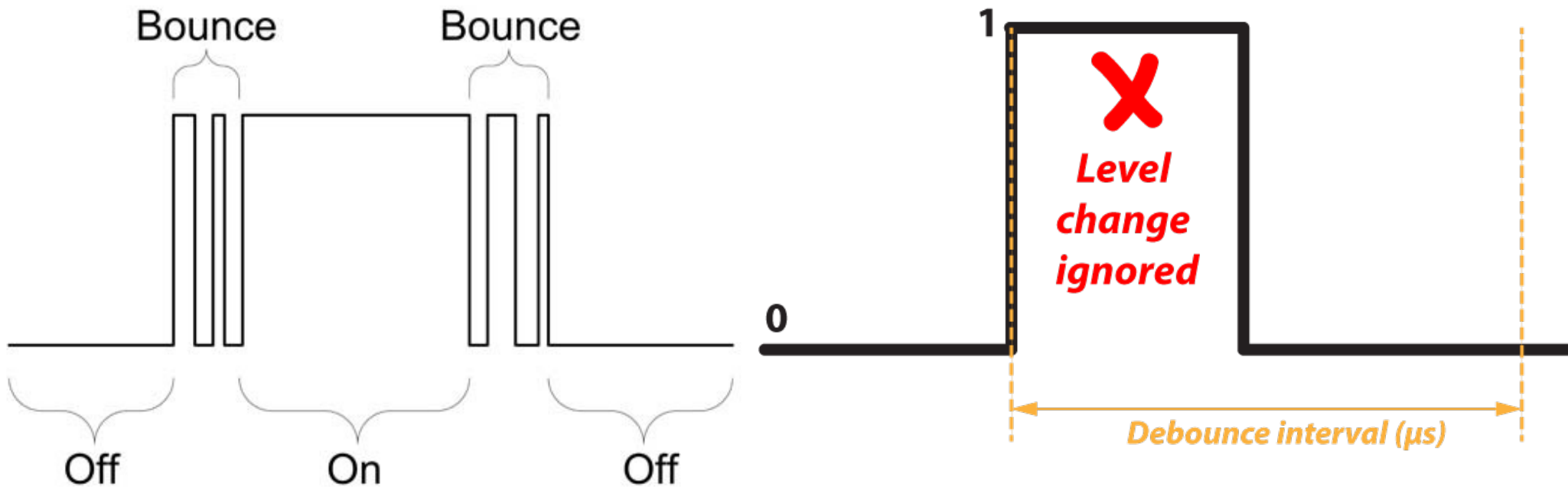
Debounce del pulsante

- Possiamo ridurre le possibilità di registrare falsi segnali con tecniche di debounce:



Debounce del pulsante

- Ritardiamo per un brevissimo intervallo di tempo l'esecuzione del codice legato alla pressione del pulsante



Debounce del pulsante

```
const int debounceDelay = 50;  
// l'assestamento in millisecondi
```

```
if (ledState == HIGH) {  
  delay(debounceDelay);  
  digitalWrite(ledPin, HIGH);  
  Serial.println("on");  
}
```