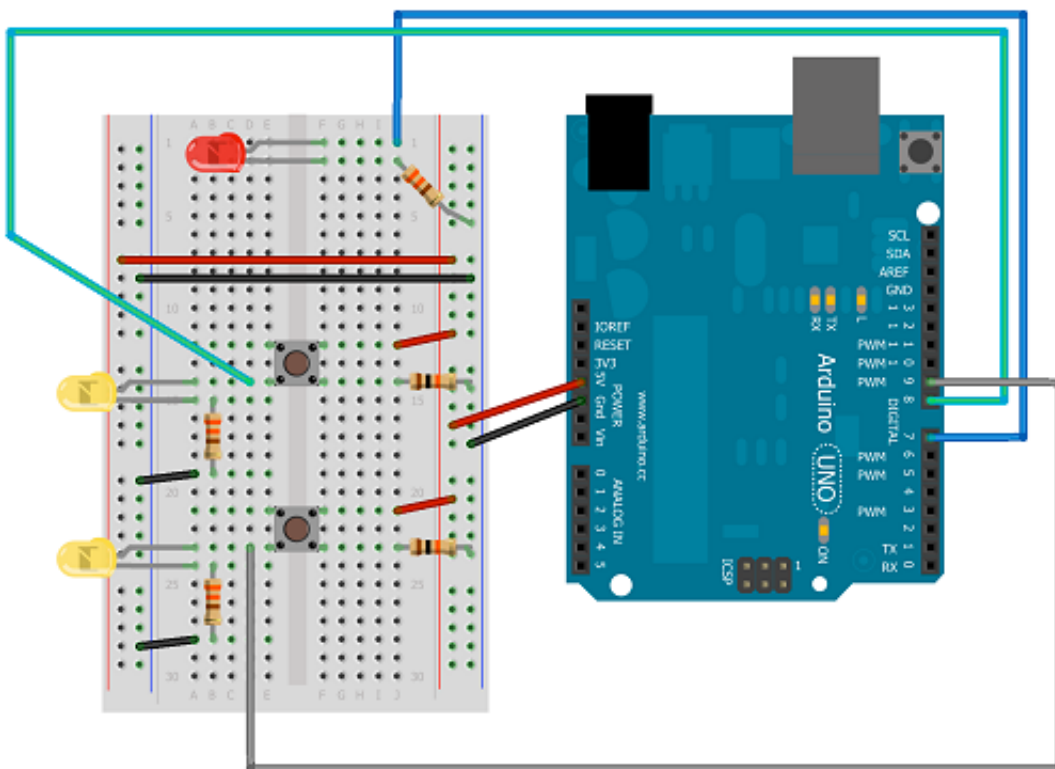
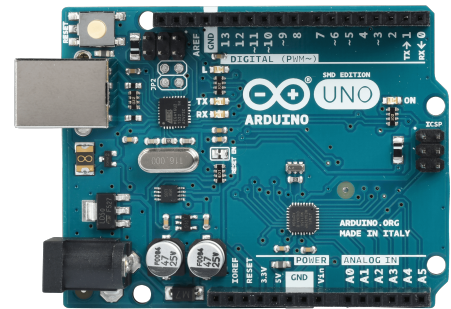
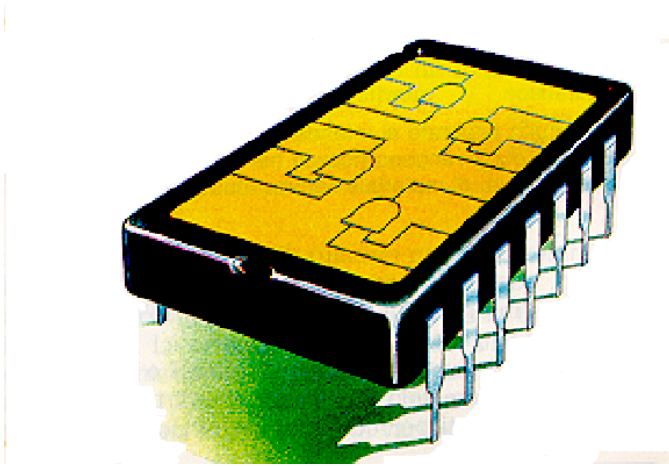


Lez. N. 06 LOGIC GATES



Materiali

Arduino

2 Led gialli + 2 Led rossi + 2 pulsanti

2 resistenze da 330 Ω e 2 resistenze da 10 K Ω

Jumper vari

Introduzione

Questo progetto è un modo semplice di usare Arduino per simulare il comportamento delle porte logiche.

Il progetto non svolge effettivamente la funzione del gate logico, ma attiva o disattiva una luce in base a uno o due ingressi. Mostra in modo efficace la tabella di verità per un determinato gate logico.

I pulsanti sono collegati con resistori da 10K a GND. Un LED è posto sulla stessa riga delle connessioni ai pin di ingresso (8 e 9). Questo ci dà una configurazione push-to-make per i pulsanti. Significa anche che non dobbiamo scrivere alcun codice per accendere i due LED indicatori di ingresso. È possibile testare le connessioni dei pulsanti in anticipo. I LED gialli in questo diagramma dovrebbero accendersi quando i pulsanti sono premuti e spenti quando non lo sono.

Codice

```
int pinOut = 7;
int pinA = 8;
int pinB = 9;

void setup()
{
  pinMode(pinOut, OUTPUT);
  pinMode(pinA, INPUT);
  pinMode(pinB, INPUT);
}

void loop()
{
  boolean pinAState = digitalRead(pinA);
  boolean pinBState = digitalRead(pinB);
  boolean pinOutState;
  // and
  pinOutState = pinAState & pinBState;
  digitalWrite(pinOut, pinOutState);
}
```

Questo è il codice di base per dimostrare il comportamento di una porta AND. Una volta caricato, dovresti essere in grado di vedere la luce del LED di uscita quando, e solo quando, entrambi i pulsanti vengono premuti allo stesso tempo. Si noti che i valori booleani vengono utilizzati qui.

Se si sostituisce la linea, **pinOutState = pinAState e pinBState;** con una delle seguenti linee, è possibile simulare le altre porte logiche comuni. Per il cancello NOT, hai solo un singolo input.

```
//or
```

```
pinOutState = pinAState | pinBState;
```

```
// not
```

```
pinOutState = !pinAState;
```

```
// XOR
```

```
pinOutState = pinAState ^ pinBState;
```

```
//NAND
```

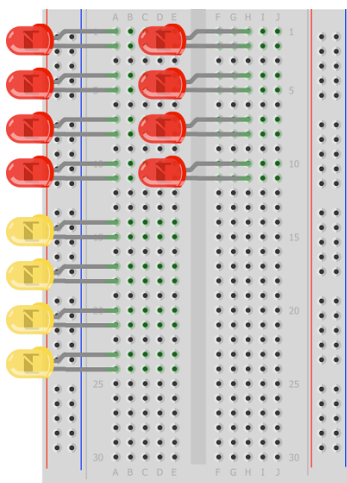
```
pinOutState = !(pinAState & pinBState);
```

```
//NOR
```

```
pinOutState = !(pinAState | pinBState);
```

Sfida

La sfida per questo circuito è di fornire un modo molto più interessante per visualizzare le informazioni. Un modo potrebbe essere quello di eliminare i pulsanti per l'input e visualizzare una tabella di verità. L'immagine seguente mostra come si possono disporre i LED,



I LED rossi sarebbero gli input, il giallo le uscite. La colonna sinistra dei LED rossi rappresenterebbe lo stato dell'ingresso A, l'ingresso della colonna della mano destra B. Qui non sono mostrati resistori, ma ne occorrerebbe uno per LED che può essere acceso. Alcuni dei LED rossi non si accenderanno mai: questi rimarrebbero semplicemente non connessi. Nessuno dei LED rossi avrebbe bisogno di essere collegato ai pin Arduino. Basta collegare quelli che rappresentano da 1 a 5 V sull'anodo con un resistore dal catodo a GND.

Ciascuno dei LED gialli richiede un resistore e una connessione a un diverso pin Arduino. Se riesci a deporre le mani su un'altra breadboard, puoi mettere i LED gialli di output su quella breadboard e allinearla per creare un buon tavolo.

Questo mostrerebbe la tabella di verità completa per una singola porta logica. Sarebbe bello avere un modo per passare tra le funzioni logiche usando i componenti. Un uso intelligente di un pulsante consente di spostarsi tra le funzioni logiche.