

Corso Arduino 2014

21 Maggio 2014

Riccardo Binetti
punkerbino@gmail.com

Lezione 1 - Blinking LEDs with style





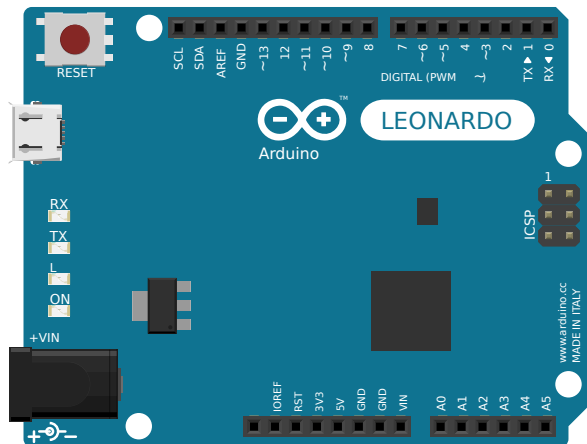
Nel dubbio, non fatelo



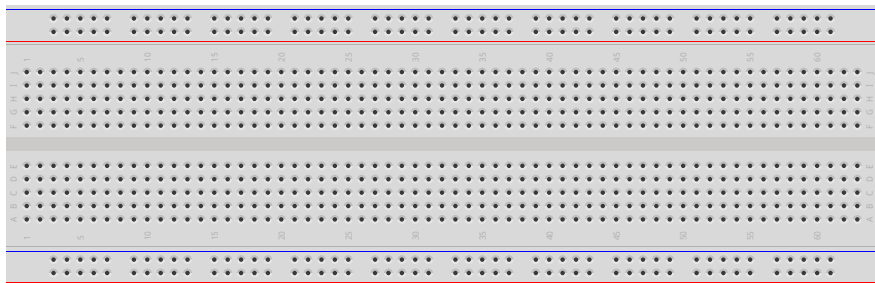


<http://bit.ly/arduino2014>

Arduino Leonardo



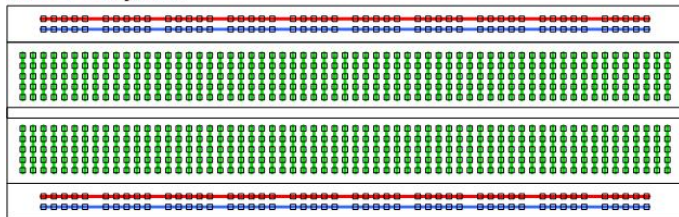
- Sarà il cervello dei nostri progetti
- Leonardo rispetto a Uno permette di essere vista come un diverso dispositivo USB



- Serve per connettere i vari componenti tra di loro
- Non richiede saldatura
- Ce ne sono di varie misure e tipi

Collegamenti della Breadboard

BPS BusBoard
Prototype
Systems

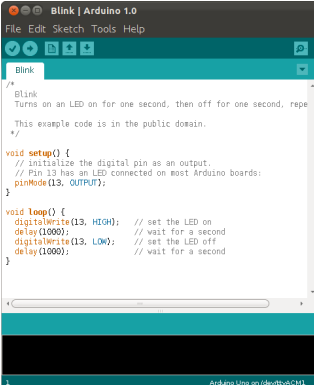


BB830 BreadBoard — Circuit Connections
Internal Connections — Distribution Strips

- I due blocchi centrali sono collegati tra di loro a blocchi di 5 (lato corto)
- I due blocchi centrali non sono collegati tra di loro
- Le 4 (o 2) file sul lato lungo sono tutte collegate tra di loro
- Su alcune breadboard, le file sul lato lungo sono interrotte a metà



- Servono a connettere Arduino alla Breadboard e i componenti quando la breadboard non basta
- Esistono dei set pretagliati per le breadboard se siete pigri
- Altrimenti basta del cavo rigido e un paio di pinze



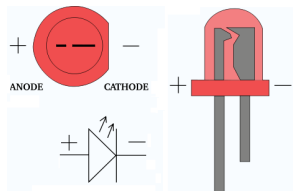
The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and other functions. The main editor area displays the following code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);           // wait for a second  
}
```

At the bottom of the window, it says "Arduino Uno on /dev/ttyCH1".

- È il programma che ci permette di scrivere sketch Arduino e di caricarli
- Contiene anche molti esempi già pronti
- Gira su Linux, Mac e Windows

Il primo componente: il LED



- Il LED si accende quando l'anodo si trova almeno a ~ 0.7 V in più rispetto al catodo
- In soldoni, il - sarà collegato a GND e il + sarà collegato in serie ad una resistenza, che poi andrà al pin
- Quasi tutte le schede Arduino hanno un LED integrato (nel caso di Arduino Leonardo è sul pin 13), per oggi useremo quello

Attenzione

Non collegate mai un LED senza una resistenza in serie

```
int ledPin = 13;
void setup(){
  pinMode(ledPin, OUTPUT);
}
void loop(){
  digitalWrite(ledPin, HIGH);
}
```

Il risultato dovrebbe essere il LED acceso fisso

Attenzione

Prima di caricare il codice, selezionate la board Arduino Leonardo
(Strumenti -> Tipo di Arduino)

```
int ledPin = 13;
```

- Assegna alla variabile ledPin il numero 13
- Non è indispensabile ma appena i pin usati superano i 2 diventa comodo
- Va definito fuori dalle funzioni di setup e loop (variabile globale)

```
void setup() {}
```

- È una delle due funzioni che devono essere definite in ogni sketch
- Tutto ciò che è al suo interno viene eseguito solo una volta appena Arduino viene accesa
- Di solito la si usa appunto per il setup iniziale

```
void loop(){}
```

- L'altra funzione che deve venire definita in ogni sketch
- Come si intuisce dal nome, viene eseguita tutta e poi si rinizia da capo, finchè non si spegne Arduino

```
pinMode(ledPin, OUTPUT);
```

- Imposta la modalità del pin
- Il primo argomento è il numero del pin che si vuole impostare
- Il secondo è la modalità (INPUT, OUTPUT o INPUT_PULLUP)

```
digitalWrite(ledPin, HIGH);
```

- Setta un'uscita digitale
- Il primo argomento è il numero del pin che si vuole settare
- Il secondo è lo stato che si vuole impostare (HIGH o LOW)
- Su Arduino Leonardo, HIGH significa che il pin sarà a 5v, LOW che sarà a 0v

```
delay(int nMilliseconds);
```

- Interrompe l'esecuzione per nMilliseconds

Con questa funzione dovrete ora essere in grado di scrivere l'Hello World per eccellenza di Arduino: il blinking LED. Dovete fare accendere e spegnere il LED ogni secondo.

Go!

```
int ledPin = 13;
int interval = 1000;
void setup(){
  pinMode(ledPin, OUTPUT);
}
void loop(){
  digitalWrite(ledPin, HIGH);
  delay(interval);
  digitalWrite(ledPin, LOW);
  delay(interval);
}
```

Finora abbiamo usato solo i pin come output. Come avrete intuito possono essere usati anche come input.

Come impostiamo il pin?


```
pinMode(ledPin, INPUT);  
digitalRead(pin);
```

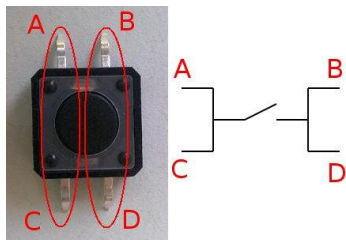
- Legge il pin (indicato con un numero) e ritorna HIGH o LOW

```
if (condizione) {...} else {...}
```

- se condizione è true esegue il codice tra la prima coppia di parentesi graffe, altrimenti il codice tra la seconda coppia di parentesi graffe (non è obbligatorio avere l'else)

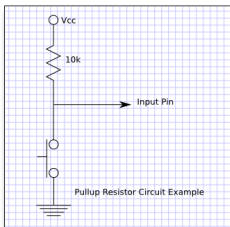
```
A == B
```

- Se A è uguale a B, ritorna true, altrimenti false



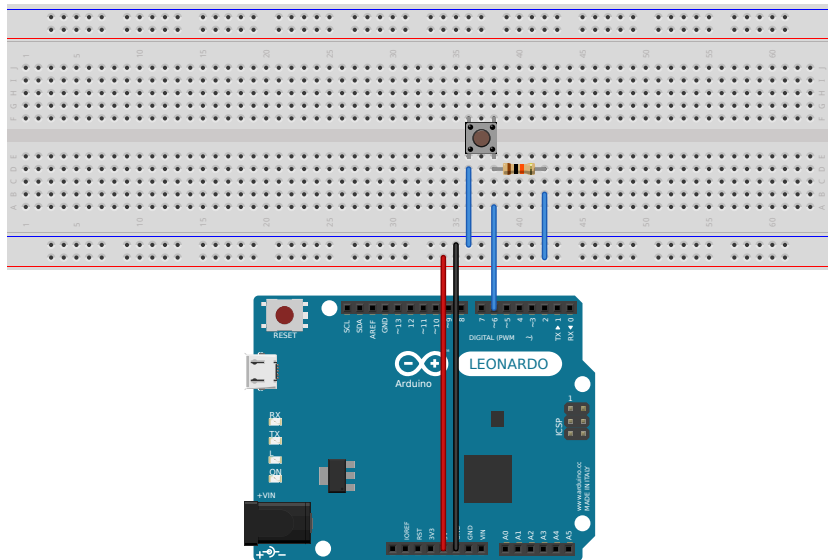
- A e C sono sempre connessi, B e D sono sempre connessi. Quando si schiaccia il bottone sono tutti connessi
- Cercate sempre informazioni sul bottone che state usando per esserne sicuri

Resistenza di pull-up (o pull-down)



Quando l'input pin non è connesso a niente, il suo ingresso è "floating", non ha un valore ben definito. Aggiungendo una resistenza cosiddetta di pull-up, quando il bottone è aperto l'input pin viene "tirato" a Vcc (quindi a HIGH), mentre quando si schiaccia il bottone l'input Pin va a LOW. Facendo la stessa cosa invertendo il sopra con il sotto del circuito si ha una resistenza di pull-down.

Circuito bottone



Realizzate un blinking LED che blinka soltanto quando il bottone è premuto, mentre rimane spento se il bottone viene rilasciato.

Go!

```
int ledPin = 13;
int buttonPin = 6;
int interval = 1000;
void setup(){
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
void loop(){
  if (digitalRead(buttonPin) == LOW){
    digitalWrite(ledPin, HIGH);
    delay(interval);
    digitalWrite(ledPin, LOW);
    delay(interval);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Domande?

Se vi vengono in mente più tardi, fatele sul gruppo

Ci vediamo settimana prossima



Queste slides sono licenziate Creative Commons Attribution-ShareAlike 3.0 Unported

<http://www.poul.org>