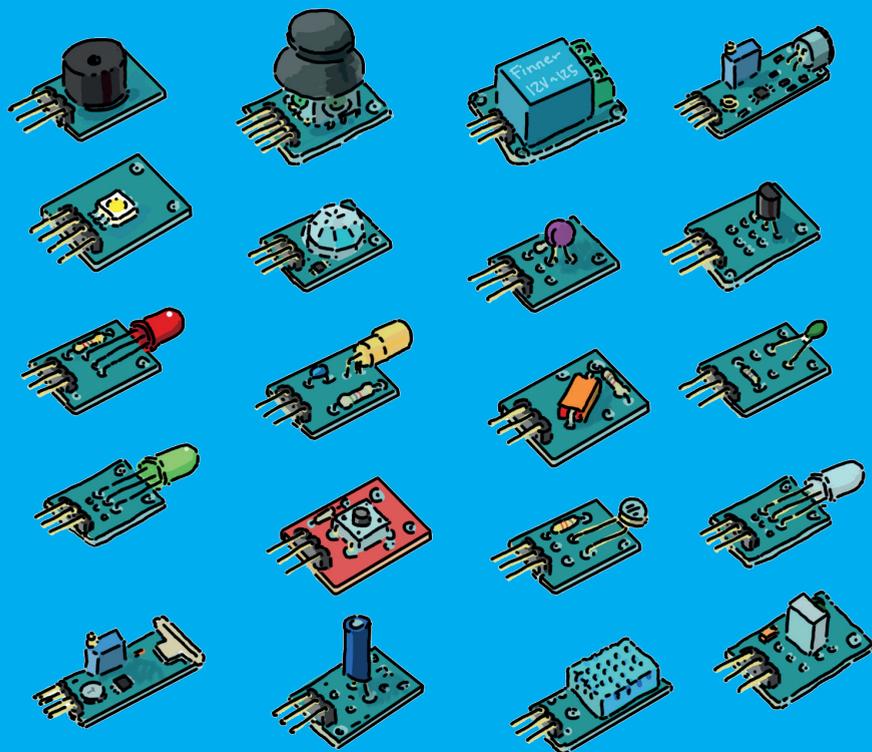


Paolo Aliverti

Arduino trucchi e segreti

120 idee per risolvere ogni problema



Programmazione: cicli, array, test e algoritmi >>

Elettronica: transistor, motori, sensori, display >>

Schede esterne: shield e breakout board >>

Comunicazione: Wi-Fi, RS232, Bluetooth >>

MADE FOR
MAKERS

EDIZIONI
LSWR



Arduino

trucchi e segreti

**120 idee
per risolvere ogni problema**

Paolo Aliverti

EDIZIONI
LSWR

Arduino trucchi e segreti

Autore: Paolo Aliverti

Collana:



Publisher: Marco Aleotti

Progetto grafico: Roberta Venturieri

Immagine di copertina: Paolo Aliverti

Logo design: Giampiero Carella

© 2018 Edizioni Lswr* - Tutti i diritti riservati

ISBN: 978-88-6895-681-3

I diritti di traduzione, di memorizzazione elettronica, di riproduzione e adattamento totale o parziale con qualsiasi mezzo (compresi i microfilm e le copie fotostatiche), sono riservati per tutti i Paesi. Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633.

Le fotocopie effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate a seguito di specifica autorizzazione rilasciata da CLEARedi, Centro Licenze e Autorizzazioni per le Riproduzioni Editoriali, Corso di Porta Romana 108, 20122 Milano, e-mail autorizzazioni@clearedi.org e sito web www.clearedi.org. La presente pubblicazione contiene le opinioni dell'autore e ha lo scopo di fornire informazioni precise e accurate. L'elaborazione dei testi, anche se curata con scrupolosa attenzione, non può comportare specifiche responsabilità in capo all'autore e/o all'editore per eventuali errori o inesattezze.

L'Editore ha compiuto ogni sforzo per ottenere e citare le fonti esatte delle illustrazioni. Qualora in qualche caso non fosse riuscito a reperire gli aventi diritto è a disposizione per rimediare a eventuali involontarie omissioni o errori nei riferimenti citati.

Tutti i marchi registrati citati appartengono ai legittimi proprietari.

All rights reserved. This translation published under license.

EDIZIONI
LSWR

Via G. Spadolini, 7
20141 Milano (MI)
Tel. 02 881841
www.edizionilswr.it

Printed in Italy

Finito di stampare nel mese di ottobre 2018 presso "Rotomail Italia" S.p.A., Vignate (MI)

(*) Edizioni Lswr è un marchio di La Tribuna Srl. La Tribuna Srl fa parte di LSWR GROUP.

Sommario

| | | |
|----|--|-----|
| 1. | PER INIZIARE | 9 |
| | 1. Installare Arduino | 10 |
| | 2. Caricare uno sketch su Arduino | 13 |
| | 3. Programmare Arduino con Arduino Web Editor | 15 |
| | 4. Caricare uno sketch con Arduino Web Editor | 17 |
| | 5. Simulare Arduino con TinkerCAD Circuits | 19 |
| | 6. Programmare Arduino a blocchi con Bitbloq | 23 |
| 2. | PROGRAMMARE ARDUINO | 29 |
| | 7. Creare uno sketch vuoto | 30 |
| | 8. Utilizzare una variabile | 31 |
| | 9. Definire una costante | 37 |
| | 10. Manipolare stringhe di caratteri | 39 |
| | 11. Usare le stringhe | 42 |
| | 12. Convertire una variabile da un tipo all'altro | 45 |
| | 13. Ripetere un'operazione | 49 |
| | 14. Utilizzare un array | 55 |
| | 15. Ottenere la lunghezza di un array | 56 |
| | 16. Copiare o estendere un array | 58 |
| | 17. Utilizzare array bidimensionali (matrici) | 61 |
| | 18. Definire un test semplice | 63 |
| | 19. Definire un test con più alternative | 68 |
| | 20. Definire una funzione | 70 |
| | 21. Creare classi e oggetti | 75 |
| | 22. Generare numeri casuali | 81 |
| | 23. Trovare il numero più grande in un array | 82 |
| | 24. Ordinare un array di numeri | 85 |
| | 25. Riempire un array di numeri casuali | 88 |
| | 26. Trasformare una stringa in un array (tokenize) | 90 |
| | 27. Gestire indici multipli e periodici | 92 |
| | 28. Lavorare con i bit | 94 |
| | 29. Creare un modulo | 101 |

| | |
|--|------------|
| 30. Usare librerie esterne..... | 103 |
| 31. Creare proprie librerie..... | 105 |
| 3. LE BASI: LED E PULSANTI | 109 |
| 32. Accendere LED..... | 110 |
| 33. Controllare la luminosità di un LED | 113 |
| 34. Usare un LED RGB..... | 117 |
| 35. Creare un PWM su ogni pin (PWM software) | 118 |
| 36. Generare una sequenza casuale di numeri per accendere dei LED..... | 120 |
| 37. Collegare pulsanti | 123 |
| 38. Debouncing..... | 127 |
| 39. Mantenere lo stato di un pulsante | 130 |
| 40. Individuare la pressione e il rilascio di un pulsante (fronti di salita e discesa) | 135 |
| 41. Misurare il tempo di pressione di un pulsante | 138 |
| 42. Rilevare clic multipli su un pulsante | 141 |
| 43. Usare un ingresso analogico | 144 |
| 44. Pulsante su linea analogica | 146 |
| 45. Utilizzare un trimmer per impostare parametri | 148 |
| 46. Inserire pause “lunghe” | 150 |
| 47. Guadagnare sei pin digitali | 152 |
| 48. Utilizzare la console per inviare e ricevere comandi | 153 |
| 49. Accendere quattro LED in sequenza con un pulsante..... | 156 |
| 50. Convertire un valore da un intervallo all’altro..... | 159 |
| 51. Indovina il LED (programma a stati) | 162 |
| 53. Controllare il riempimento di una vasca (macchina a stati) | 169 |
| 54. Accendere una caldaia | 175 |
| 55. Simon..... | 181 |
| 56. Comporre un numero premendo più tasti | 190 |
| 4. I SENSORI | 195 |
| 57. Alimentare Arduino | 196 |
| 58. Misurare una tensione..... | 199 |
| 59. Misurare una corrente..... | 202 |
| 60. Collegare un sensore di temperatura (termistore) | 206 |
| 61. Collegare un sensore di temperatura LM35..... | 210 |
| 62. Collegare un sensore di luce | 212 |
| 63. Collegare un sensore per l’umidità | 215 |
| 64. Utilizzare un sensore digitale per umidità e temperatura | 217 |
| 65. Rilevare un gas..... | 219 |
| 66. Rilevare campi magnetici..... | 225 |
| 67. Rilevare l’inclinazione con un tilt sensor | 229 |
| 68. Rilevare il movimento con un PIR..... | 230 |
| 69. Realizzare un pulsante a sfioramento o touch sensor | 232 |
| 70. Utilizzare un sensore a infrarossi per decodificare un telecomando | 235 |
| 71. Accendere la TV con Arduino..... | 239 |

| | |
|---|-----|
| 72. Rilevare un ostacolo con un sensore a infrarossi | 242 |
| 73. Misurare la distanza con un sensore a ultrasuoni..... | 245 |
| 74. Utilizzare data e ora corrente | 249 |
| 75. Usare un tastierino numerico..... | 252 |
| 76. Rilevare l'orientamento o il moto con accelerometri e giroscopi | 256 |
| 78. Trovare il nord con una bussola | 260 |
| 79. Conoscere la posizione geografica con un GPS..... | 264 |
| 80. Collegare un microfono e realizzare un "clap control" | 270 |
| 81. Collegare una fonte audio..... | 273 |
| 82. Leggere tag RFID | 276 |
| 83. Isolare un ingresso | 280 |
| | |
| 5. GLI ATTUATORI E LA VISUALIZZAZIONE DI INFORMAZIONI | 283 |
| 84. Accendere una lampada a 12 V con un transistor | 284 |
| 85. Pilotare un relè con un transistor | 286 |
| 86. Accendere una lampada a 12 V con un MOSFET | 289 |
| 87. Controllare la velocità di un motore..... | 292 |
| 88. Pilotare la direzione di rotazione di un motore c.c. | 295 |
| 89. Utilizzare un servomotore..... | 298 |
| 90. Accendere una lampada a 220V | 302 |
| 91. Interruttore crepuscolare per una lampada a 220 V | 304 |
| 92. Usare un motore passo-passo (stepper) | 305 |
| 93. Isolare le uscite | 309 |
| 94. Aumentare il numero di out con uno shift register | 312 |
| 95. Utilizzare un display a sette segmenti | 318 |
| 96. Usare un display LCD..... | 322 |
| 97. Comandare un display con soli 2 fili (I ² C)..... | 327 |
| 98. Disegnare caratteri custom su un display LCD | 329 |
| 99. Realizzare un LCD ticker..... | 330 |
| 100. Realizzare un menu multilivello con display LCD | 334 |
| 101. Collegare un display grafico TFT | 336 |
| 102. Generare suoni | 339 |
| 103. NeoPixel LED | 342 |
| 104. Generare un segnale analogico | 345 |
| 105. Collegare un amplificatore..... | 348 |
| 106. Controllare un potenziometro digitale..... | 350 |
| 107. Riprodurre un file audio mp3 o wav | 355 |
| | |
| 6. COMUNICARE E SCAMBIARE DATI | 361 |
| 108. Aggiungere una nuova porta seriale (SoftwareSerial) | 362 |
| 109. Interfacciarsi a una porta RS232 | 364 |
| 110. Interfacciarsi a una porta RS485..... | 367 |
| 111. Far comunicare due Arduino con il protocollo I ² C | 372 |
| 112. Far comunicare due Arduino con il protocollo SPI | 375 |
| 113. Controllare Arduino con Firmata..... | 380 |

| | |
|--|-----|
| 114. Usare Arduino come console per Processing | 385 |
| 115. Controllare un LED via Bluetooth con uno smartphone | 387 |
| 116. Accendere un LED con Python..... | 392 |
| 117. Trasmettere dati da Arduino a Python | 395 |
| 118. Connettersi a Internet e leggere una pagina web | 397 |
| 119. Arduino come web server | 402 |
| 120. Pubblicare una pagina web dinamica..... | 407 |
| 121. Accendere un LED con il browser..... | 413 |
| 122. Inviare una mail con Arduino premendo un tasto | 417 |
| 123. Inviare SMS con Arduino | 423 |
| 124. Ricevere SMS | 426 |
| 125. Leggere e scrivere un file da SD Card | 428 |
| 126. Generare un segnale TV | 431 |
| 127. Emulare un mouse USB | 432 |
| 128. Emulare una tastiera USB..... | 435 |
| | |
| 7. INDICE ANALITICO | 437 |

Per iniziare

In questa sezione troverai le risposte ai più comuni problemi di **installazione e configurazione di Arduino**. Vedrai anche varie **alternative per scrivere degli sketch e caricarli sulla scheda, scrivendo codice o componendo blocchi**.

Le prime volte che ci si avvicina a un nuovo prodotto o a una nuova tecnologia ci sembra tutto difficile e oscuro. Anche Arduino, per quanto possa essere semplice, può presentare delle difficoltà iniziali. I problemi maggiori si incontrano nella fase di installazione, anche se con il tempo il processo di `setup()` è stato migliorato e ora è raro incappare in qualche tipo di problema. Con le prime schede come Arduino Diecimila o Duemilanove c'era sempre il dramma dell'installazione dei driver per la porta seriale, oggi caso abbastanza raro.

In questa prima sezione troverete alcuni "trucchi" per risolvere i problemi che potreste incontrare in fase di installazione e configurazione del prodotto. Il metodo classico per programmare Arduino è quello di utilizzare il suo ambiente di sviluppo (IDE). Oggi è possibile programmare Arduino anche con piattaforme "in cloud" (online) e persino simularlo, così che la maggior parte degli sketch e degli esempi si possono provare anche senza disporre di una scheda reale.

Il grande successo che ha avuto Scratch (<https://scratch.mit.edu>), un sistema sviluppato dal MIT per scrivere codice componendo elementi grafici, è arrivato anche ad Arduino, definendo quasi uno standard per questo tipo di interfacce di programmazione. Per questo si trovano sistemi come TinkerCAD Circuits e Bitbloq con cui programmare Arduino a blocchi. Con questi approcci è difficile scrivere codice comples-

so, ma sono molto efficaci per far avvicinare con successo bambini e principianti alla programmazione.

1. Installare Arduino

Scaricate Arduino IDE dal sito <http://www.arduino.cc/en/Main/Software> e seguite le istruzioni fornite per il vostro sistema (Windows, Mac o Linux). Arduino IDE è un programma che dovete installare sul vostro computer per poter programmare Arduino.



Figura 1.1 - Download del software per Arduino.

In dettaglio

Visitate l'indirizzo <http://www.arduino.cc/en/Main/Software> e scaricate l'ultima versione del software. Nel momento in cui scrivo la versione pubblicata è la 1.8.5 ed è disponibile per Windows, Mac OS X e Linux. Scegliete la versione compatibile con il vostro sistema e scaricatela in una cartella.

Windows

La procedura di installazione di Arduino è in genere abbastanza semplice.

Se utilizzate Windows potete far riferimento alle istruzioni pubblicate sul sito di Arduino (<http://www.arduino.cc/en/Guide/Windows>).

Per iniziare procuratevi una scheda Arduino e un cavo per collegarla al vostro computer. Il cavo dipende dalla versione di Arduino che avete acquistato. Per la scheda più diffusa, Arduino Uno, vi servirà un cavo USB tipo A-B, di quelli che si utilizzano per le

stampanti (Figura 1.2). Il connettore USB maschio tipo B è di forma rettangolare, mentre il connettore maschio di tipo A ha una forma quadrata, con due spigoli smussati. Se avete un Arduino Mega, un Due o un Leonardo, dovrete invece utilizzare un cavo micro-USB: il cavo utilizzato di solito per caricare i cellulari Android.



Figura 1.2 – Cavo USB di tipo A-B, adatto per Arduino Uno.

Se la scheda è nuova infilate il cavo e fate attenzione che il connettore entri bene nell'alloggiamento. Quando utilizzate i cavi micro-USB, fate molta attenzione perché, anche se la presa micro-USB è saldata sul circuito stampato, è molto facile spezzarla con una piccola flessione.

Collegate Arduino al vostro computer: la scheda dovrebbe accendersi. Tutte le schede Arduino sono dotate di almeno un LED che inizierà a lampeggiare o comunque a indicare qualche tipo di attività. Probabilmente Windows rileverà¹ la presenza di un nuovo hardware e segnalerà la ricerca o l'installazione di nuovi driver. Con le ultime versioni del software, la procedura di installazione si è notevolmente semplificata ed è veramente raro dover installare manualmente i driver, se non nel caso di alcune schede particolari. La descrizione ufficiale della procedura di installazione, riportata sul sito di Arduino dice:

1. collegate Arduino al computer e attendete che Windows inizi il processo di installazione dei driver fino a che l'installazione fallirà;
2. aprite il Pannello di controllo, quindi cliccate su "Sistema e sicurezza" e infine su "Sistema";
3. nella finestra "Sistema", aprite "Gestione dispositivi";

¹ Il condizionale è d'obbligo vista l'eterogenea varietà di configurazioni hardware e versioni di Windows.

4. nell'elenco dell'hardware cercate la voce "Porte seriali e parallele" (COM e LPT). Tra le voci dovrete trovare una porta "Arduino Uno (COMxx)". Se non trovate la periferica, allora controllate in "Altri dispositivi" o "Dispositivi sconosciuti";
5. cliccate con il tasto destro del mouse su "Arduino Uno (COMxx)" e scegliete "Aggiorna il software del driver";
6. scegliete "Sfogliate le cartelle per cercare il driver";
7. navigate tra le cartelle dove avete scaricato e spaccettato il software di Arduino e localizzate il file `arduino.inf` che troverete nella cartella "Driver" (non nella cartella "FTDI USB Drivers"!);. Nelle vecchie versioni di Arduino IDE (fino alla 1.0.3) troverete un file chiamato `Arduino Uno.inf`;
8. Windows riconoscerà il driver e l'installazione si concluderà.

Ora potete avviare Arduino cliccando sull'icona apparsa sul Desktop o nel menu di avvio.

Mac OS X

Per OS X la procedura di installazione è ancora più semplice. Potete far riferimento alle istruzioni pubblicate sul sito di Arduino (<http://www.arduino.cc/en/Guide/MacOSX>). Dovreste aver scaricato un file compresso (.zip) sul vostro computer. Solitamente il file finisce nella cartella "Download" o sul Desktop. Scompattate il file con un doppio clic. Vedrete apparire un file `Arduino.app`: trascinatelo nella cartella "Applicazioni".

Con le nuove schede in genere non avrete bisogno di fare altro, non dovrete installare nessun tipo di driver. I driver FTDI vi serviranno se per caso vorrete utilizzare una scheda "Duemilanove". Trovate i driver sul sito FTDI (<http://www.ftdichip.com/Drivers/VCP.htm>).

Avviate Arduino con un doppio clic sull'icona che trovate nella cartella "Applicazioni".

GNU/Linux

Dalla versione dieci di Ubuntu, troverete Arduino nei "repository" di sistema ("Universe"). Utilizzate synaptic per individuare il pacchetto e installarlo. In alternativa aprite un terminale e digitate:

```
sudo apt-get update && sudo apt-get install arduino arduino-core
```

Dovrete inserire la vostra password perché state utilizzando il comando `sudo`. Al termine dell'installazione troverete Arduino tra i programmi installati.

Se il software non dovesse avviarsi, potrebbe essere necessario installare Java. Installate il pacchetto `open-jdk` utilizzando synaptic o da terminale.

Verificate che il vostro utente sia nel gruppo "dialout", così che abbia i permessi per utilizzare le porte seriali. Se così non fosse (o in caso di dubbio) aggiungete il gruppo con:

```
sudo usermod -aG dialout paolo
```

Sostituite `paolo` con il nome del vostro utente. In alternativa potete lanciare Arduino da terminale usando `sudo`:

```
sudo ./arduino
```

2. Caricare uno sketch su Arduino

Per caricare uno sketch su Arduino premete il pulsante "Verify" posto sulla toolbar dell'IDE, quindi "Upload". Il caricamento si può fare direttamente premendo solo "Upload".

In dettaglio

Lavorando con Arduino dovreste caricare nuovi sketch in continuazione sulla scheda. L'operazione di caricamento è molto semplice e rapida e ha caratterizzato il successo di Arduino. Alcuni anni fa la procedura di programmazione di un microcontrollore era una faccenda abbastanza complessa, mentre oggi è questione di pochi secondi. Per caricare un nuovo sketch sulla vostra scheda vi serve solo un cavo USB di tipo adatto. Il cavo dipende dal tipo di scheda e purtroppo ogni modello ne richiede uno differente! Per programmare Arduino Uno, la scheda più diffusa, serve un cavo USB di tipo A-B, di quelli solitamente usati per collegare una stampante al computer (Figura 1.2). Se è la prima volta che caricate uno sketch, dovete prima installare il software (vedete come fare leggendo il trucco n. 1).

Per caricare uno sketch su Arduino:

1. avviate l'ambiente di programmazione di Arduino;
2. realizzate il vostro prototipo e cablatelo ad Arduino;
3. scrivete lo sketch;
4. collegate Arduino al vostro computer;
5. verificate che sia selezionata la porta corretta controllando in Tools > Port. Su Windows dovrete trovare una voce tipo COMx (Arduino Uno); su Mac avrete qualcosa tipo /dev/tty.usbmodemXXX, mentre su GNU/Linux troverete /dev/ttyACM0 o /dev/ttyS0 o S1;
6. verificate che anche l'hardware impostato in Tools > Board corrisponda alla vostra scheda (per esempio Arduino Uno);

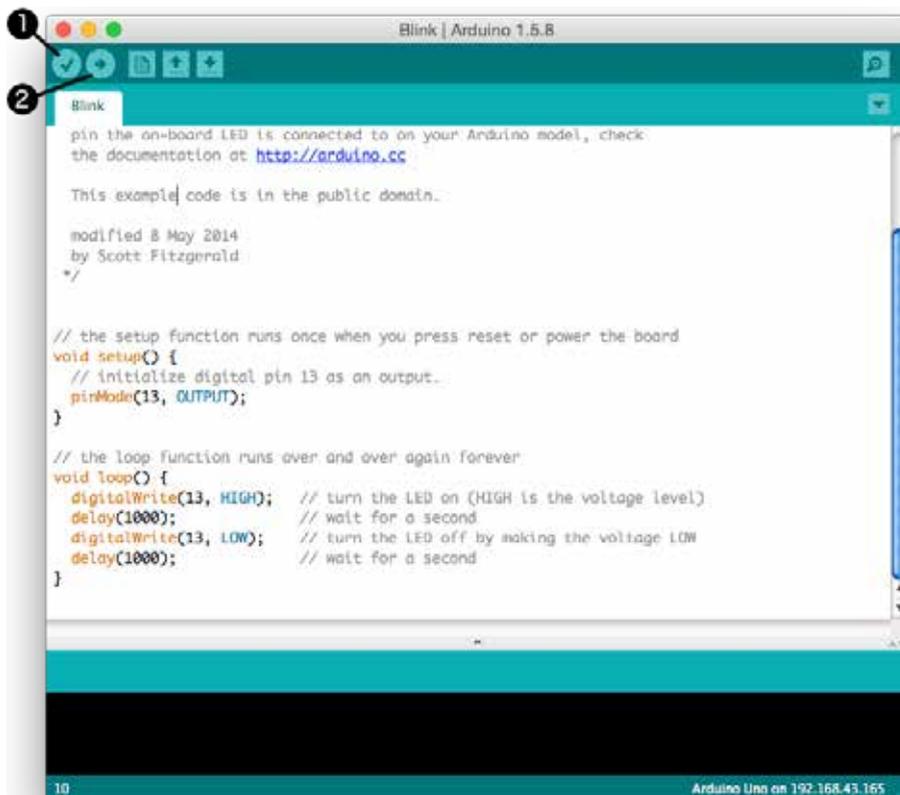
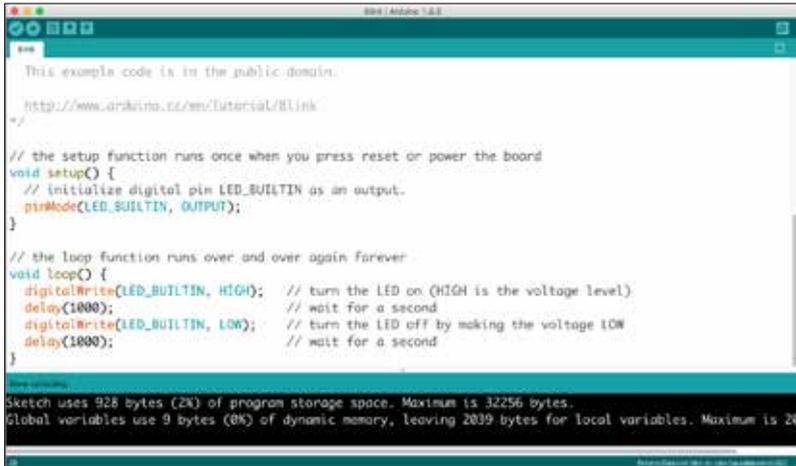


Figura 1.3 - Finestra di Arduino con caricato lo sketch Blink. Nella toolbar trovate i pulsanti "Verify" per controllare il listato (1) e "Upload" per caricarlo sulla scheda (2).

7. ora premete il pulsante "Verify" (il primo che trovate sulla toolbar);
8. il computer frullerà per qualche secondo mentre controlla il vostro listato. Se tutto è andato bene e non ci sono errori, nella parte inferiore della finestra, nella console, dovreste vedere solo delle scritte di colore bianco;
9. premete "Upload", il secondo pulsante della toolbar, per trasferire il programma alla scheda;
10. se non ci sono stati problemi di comunicazione (di solito dovuti alla scelta della porta seriale sbagliata), dopo alcuni istanti il firmware sarà trasferito e nella parte inferiore della finestra dovreste trovare la scritta "Done Uploading".

Se le cose vanno male:

- verificate la porta seriale: è quella giusta? È stata installata correttamente?
- il cavo è inserito bene nella scheda? I LED si accendono?
- avete scelto la scheda corretta?



```

This example code is in the public domain.

http://www.arduino.cc/en/tutorial/blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2839 bytes for local variables. Maximum is 28

```

Figura 1.4 - Finestra di Arduino al termine del caricamento di uno sketch.

3. Programmare Arduino con Arduino Web Editor

Arduino Web Editor è la versione online del classico Arduino IDE. Potete programmare Arduino utilizzando semplicemente un browser collegato a Internet. Per utilizzare Arduino Web Editor collegatevi al sito <https://create.arduino.cc/editor> (è necessario registrarsi).

In dettaglio

Da poco è disponibile anche una versione online, chiamata Arduino Web Editor che vi permette di programmare online e di mantenere tutti i vostri sketch “nel cloud”, cioè sui server di Arduino. Arduino Web Editor fa parte della nuova piattaforma Arduino Create, nata per aiutare i maker a scrivere codice, collaborare e trovare supporto per i propri progetti. Con questa soluzione utilizzerete sempre la versione più aggiornata dell’IDE, completa di tutte le librerie. Potete iniziare a utilizzare la versione online di Arduino visitando la pagina di download (<http://www.arduino.cc/en/Main/Software>). Per poter accedere all’editor è necessario, ovviamente, registrarsi. L’accesso avviene con una pagina dove è necessario inserire la vostra login e una password. Dopo esservi registrati, riceverete una mail con un link per attivare il vostro account e completare la procedura di registrazione. Potete accedere ad Arduino Web Editor all’indirizzo web: <https://create.arduino.cc/editor>.

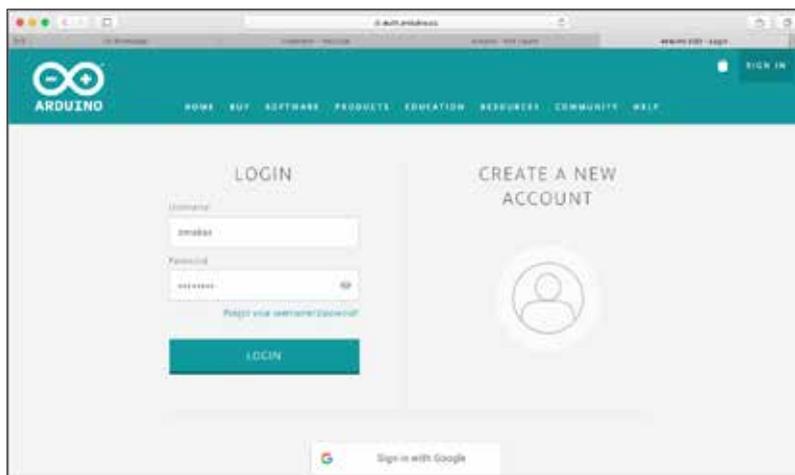


Figura 1.5 - La pagina web di accesso al Web Editor.

Per utilizzare Arduino Web Editor è comunque necessario installare un piccolo programma, chiamato "plugin" che gira sul vostro computer e comunica con la scheda Arduino e il Web Editor. Il plugin è disponibile per Windows, Linux e Mac e si installa con una procedura simile a quella di Arduino IDE. Il plugin riconosce la scheda attualmente collegata al vostro computer e gestisce il caricamento del firmware che avete scritto online.

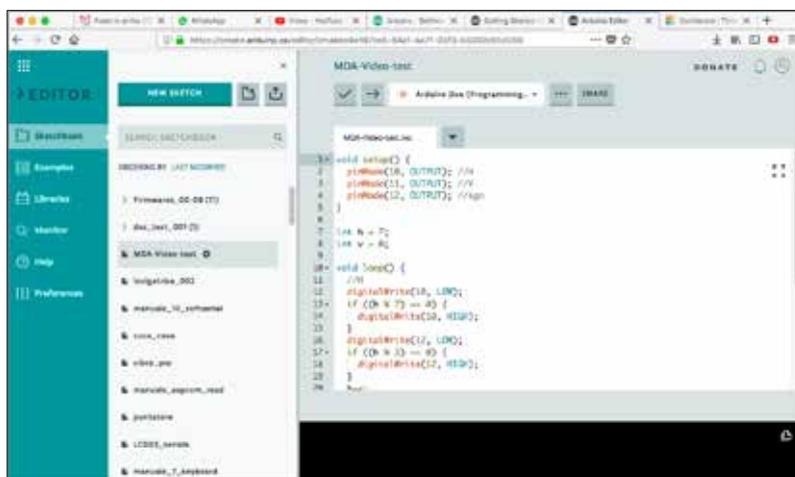


Figura 1.6 - La pagina web di accesso al Web Editor.

Il Web Editor si può utilizzare con i più comuni browser (Chrome, Firefox, Safari e Edge), anche se è consigliabile usarlo con Chrome. Una volta che vi sarete autenticati, troverete sulla sinistra, nella prima colonna, il menu principale con le voci:

- Sketchbook: il posto in cui raccoglierete tutti i vostri sketch;
- Examples: una sezione con vari esempi che potete caricare per comprendere il funzionamento di Arduino;
- Libraries: la sezione di gestione delle librerie;
- Serial Monitor: una sezione per attivare il Serial Monitor online e ricevere e trasmettere dati con l'Arduino collegato al vostro computer;
- Help: link e pagine di aiuto;
- Preferences: le opzioni di configurazione del Web Editor.

La seconda colonna riporta solitamente le opzioni relative a ogni menu, mentre la terza è l'area principale in cui scrivere il codice.

Il Web Editor è molto comodo perché vi permette di avere sempre a disposizione tutti i vostri sketch ovunque voi siate e con l'ultima versione del compilatore e delle librerie. Purtroppo potrebbe essere più lento rispetto ad Arduino IDE perché lavora online e gli sketch che realizzate vanno compilati su un server remoto e scaricati sul vostro computer.

4. Caricare uno sketch con Arduino Web Editor

Dopo aver scritto lo sketch online, all'interno di Arduino Web Editor, premete "Verify" e quindi "Upload", proprio come nell'IDE di Arduino.

In dettaglio

La procedura di upload di uno sketch con Arduino Web Editor non è tanto differente da quella classica. Per utilizzare Arduino Web Editor è necessario che vi registriate al sito e che installiate il plugin sul vostro computer (vedi trucco n. 1).

Per caricare uno sketch su Arduino Web Editor:

1. accedete ad Arduino Web Editor (<https://create.arduino.cc/editor>) con le vostre login e password;
2. realizzate il vostro prototipo e cablatelo ad Arduino;
3. scrivete lo sketch online;
4. collegate Arduino al vostro computer;
5. verificate che siano selezionate la scheda e la porta seriale corretta (Figura 1.7), utilizzando l'apposito menu a discesa che trovate all'inizio del vostro sketch;

- ora premete il pulsante "Verify" (ha un segno a forma di "V" ed è posto subito sotto al titolo dello sketch);
- premete "Upload", il pulsante di fianco a "Verify", per trasferire il programma dai server di Arduino alla vostra scheda (Figura 1.8);
- dopo alcuni istanti il firmware sarà trasferito e nella parte inferiore della finestra dovrete trovare la scritta "Success: Done Uploading" (Figura 1.9).

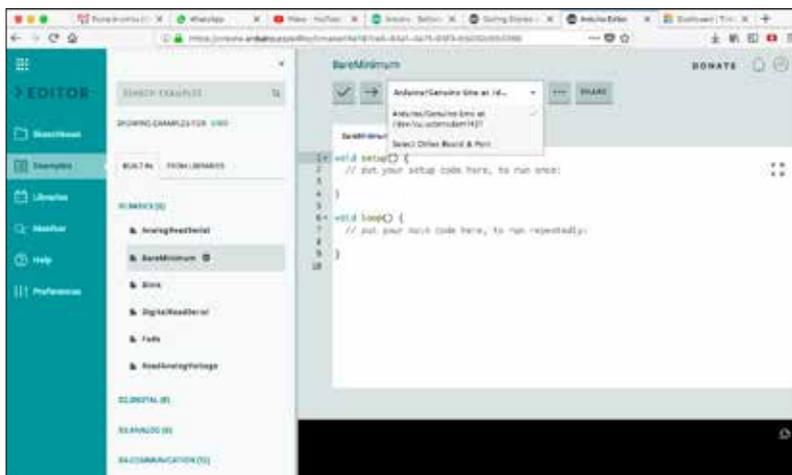


Figura 1.7 - Prima di caricare lo sketch con Arduino Web Editor, selezionate la scheda e la porta da usare.

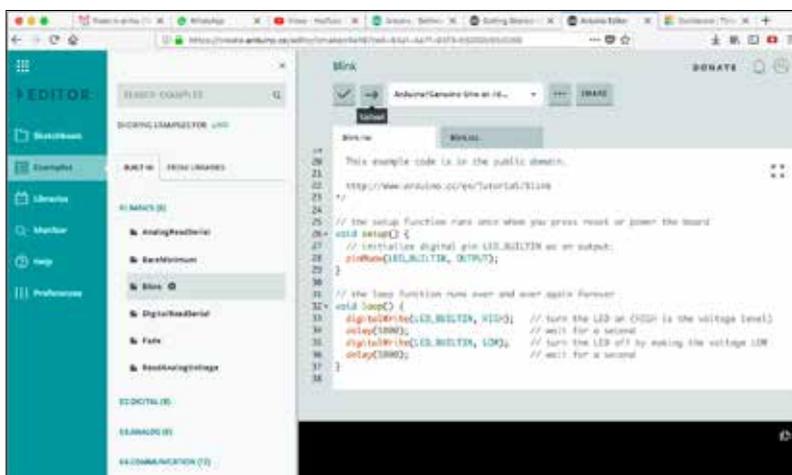


Figura 1.8 - I pulsanti di "Verify" e "Upload" si trovano poco sotto al titolo dello sketch.

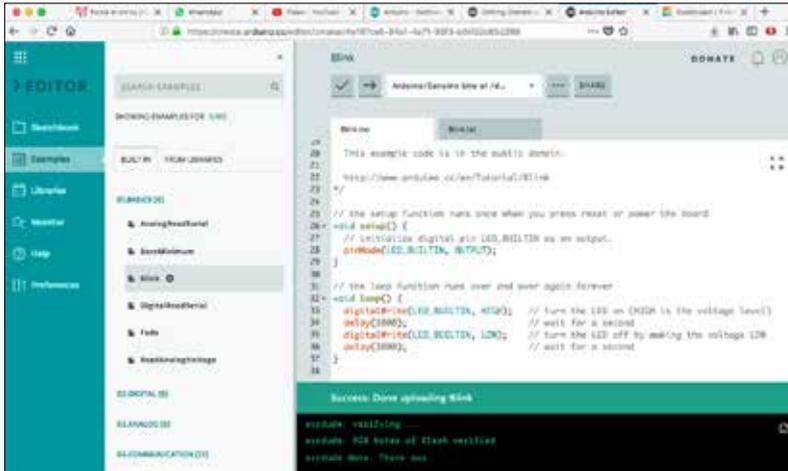


Figura 1.9 – Al termine dell'upload, nella parte inferiore della finestra troverete il messaggio: "Success: Done Uploading".

5. Simulare Arduino con TinkerCAD Circuits

È possibile assemblare e simulare un circuito elettronico con Arduino utilizzando TinkerCAD Circuits (<http://www.tinkercad.com/>), un software online nato inizialmente per la modellazione 3D, ma che da poco include anche un modulo di progettazione e simulazione elettronica. TinkerCAD Circuits richiede una registrazione per poter essere utilizzato e funziona direttamente dal browser.

In dettaglio

Esiste un simulatore per Arduino? In molti si facevano questa domanda e in effetti qualche software era disponibile da alcuni anni, ma le varie soluzioni presenti avevano tutte delle limitazioni o qualche difetto di troppo per essere considerate utilizzabili. Da poco tempo Autodesk ha approntato un modulo online per realizzare e simulare ogni tipo di circuito elettronico.

I simulatori esistono da moltissimo tempo ma sono tipicamente dei programmi ostici da utilizzare. Autodesk è riuscita a creare un programma integrato all'interno di TinkerCAD, un noto CAD online, semplicissimo da utilizzare e abbastanza affidabile. Il simulatore di Autodesk si chiama TinkerCAD Circuits e permette di simulare non solo circuiti analogici, formati con LED, resistenza, condensatori, transistor e alcuni integrati, ma di aggiungere anche Arduino, di programmarlo e debuggarlo! Una vera meraviglia.

Per utilizzare TinkerCAD Circuits è necessario registrarsi gratuitamente a TinkerCAD (<http://www.tinkercad.com/>) inserendo la propria mail e una password sicura. È anche possibile attivare un processo di login in due passi con invio di un codice numerico per l'accesso. Dopo essersi registrati è possibile accedere a TinkerCAD. La sezione Circuits si raggiunge cliccando sul link posto nella toolbar di sinistra. Nella dashboard troverete tutti i vostri Circuit, visualizzati con una preview e il titolo. Per creare un nuovo circuito cliccate su "Create new Circuit". I Circuit possono essere privati o pubblici. Per modificare la visibilità cliccate sul piccolo ingranaggio che trovate nell'angolo dell'immagine di preview e scegliete la voce "properties". Se decidete di rendere pubblico un circuito, accertatevi di completare tutte le informazioni necessarie e di inserire dei tag e una descrizione appropriata. Portando il cursore del mouse sopra all'immagine di preview appare anche la scritta "Tinker this": cliccatela per modificare il circuito.



Figura 1.10 - La dashboard di TinkerCAD Circuits.

La composizione o la modifica di un circuito sono molto semplici e immediate. L'impostazione dell'area di lavoro è molto simile a quella di Fritzing. I componenti sono raccolti nella colonna di sinistra, all'interno di un pannello dotato di alcuni filtri e di un campo di ricerca. Per creare un circuito iniziate a trascinare elementi sull'area principale. Se vi manca qualcosa, provate a cercarla. Per esempio potete richiamare una breadboard scrivendo "breadboard" nella casella di ricerca. I collegamenti elettrici si tracciano cliccando su un terminale libero e poi spostando il mouse. L'algoritmo di tracciamento dei fili è molto flessibile e permette di tracciare delle linee ordinate e con curve precise.

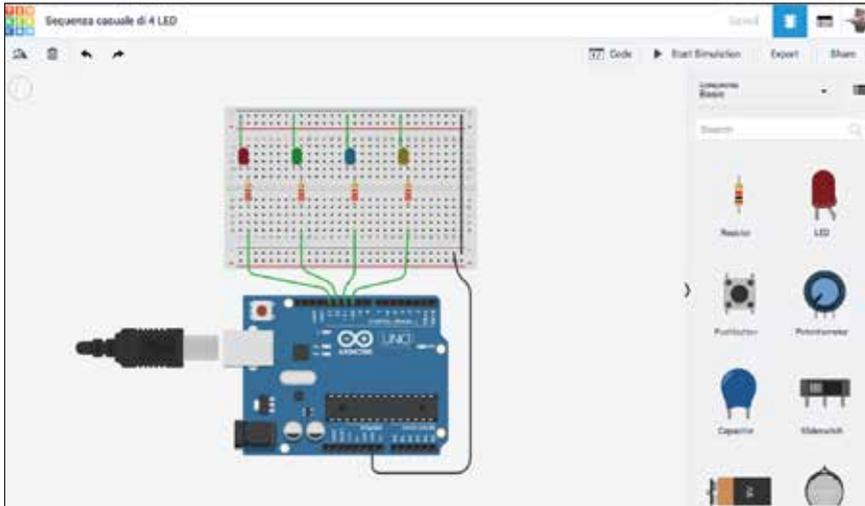


Figura 1.11 – TinkerCAD permette di creare circuiti elettronici e di utilizzare anche Arduino.

Inserendo sulla scena una scheda Arduino (per ora è disponibile solo Arduino Uno), potrete editare il codice e anche testarlo. Al codice si accede premendo il tasto “Code” posto sulla toolbar superiore. Un nuovo pannello apparirà sul lato destro e lì potrete iniziare a scrivere il codice per Arduino. La prima volta che aprirete il pannello per la scrittura del codice resterete colpiti da quello che vedrete: il codice è infatti definito a blocchi, come si fa con Scratch. Anche i più piccoli possono iniziare a sperimentare con Arduino! Per passare alla vista “classica” dello sketch, aprite il menu a discesa posto sopra al pannello che raccoglie i blocchi e scegliete “Code”.

Provate a scrivere il codice per far lampeggiare un LED:

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000); // Wait for 1000 millisecond(s)  
  digitalWrite(13, LOW);  
  delay(1000); // Wait for 1000 millisecond(s)  
}
```

Verificate che il codice sia corretto e quindi premete “Start Simulation” per avviare la simulazione. Dopo qualche istante, se non ci sono errori, il LED di Arduino inizierà a lampeggiare! Semplice no?

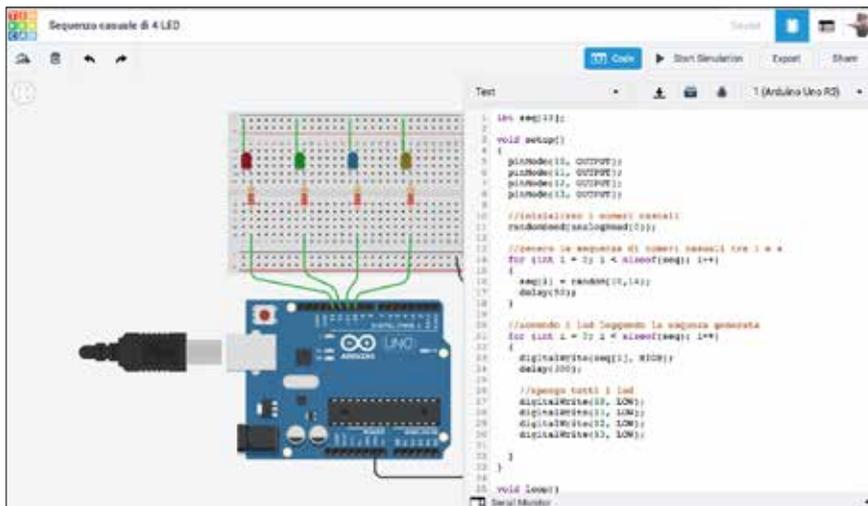


Figura 1.12 - Il codice di Arduino è accessibile premendo il pulsante "Code".

Il simulatore vi permette anche di eseguire passo per passo il codice (debug)! Inserendo dei breakpoint, cioè dei punti di stop in cui l'esecuzione del programma si fermerà, potrete analizzare i valori memorizzati nelle variabili e comprendere meglio quello che accade. Quando il programma è in debug potete farlo avanzare di breakpoint in breakpoint oppure linea per linea. Per attivare la modalità debug è sufficiente cliccare sul pulsante con la figura di uno scarafaggio stilizzato. I breakpoint si inseriscono semplicemente cliccando a inizio riga.

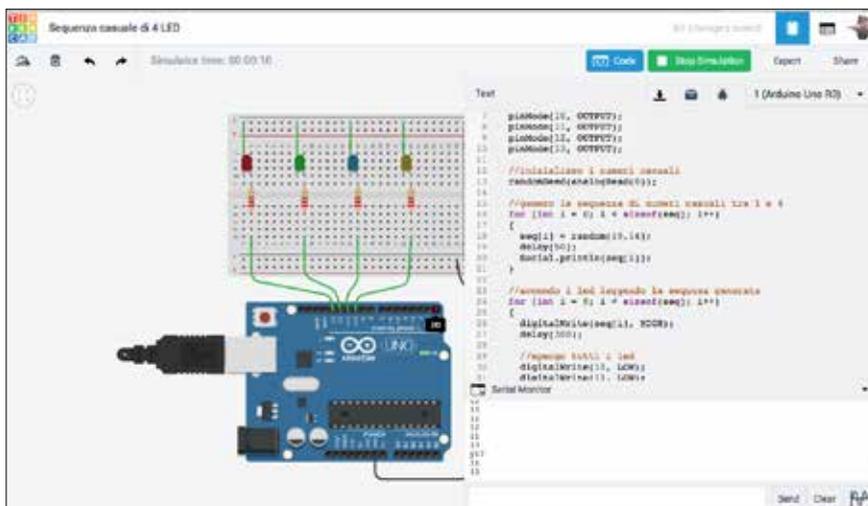


Figura 1.13 - TinkerCAD Circuits include anche il Serial Monitor e il debug.

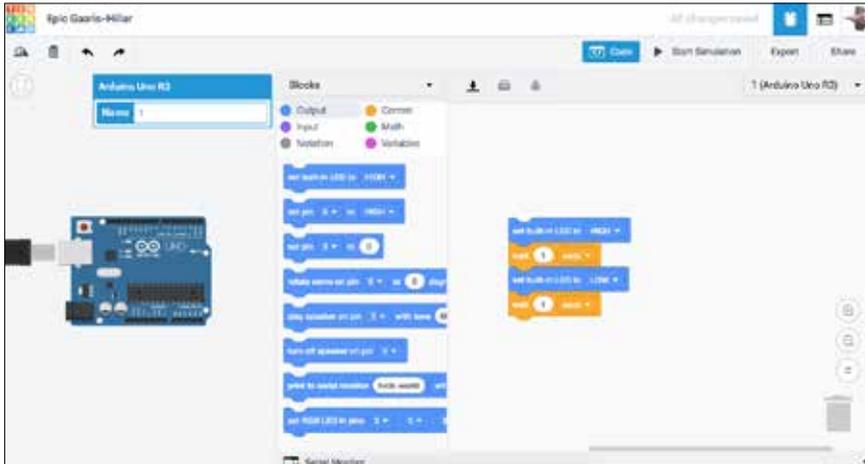


Figura 1.14 – Anche i più piccoli possono scrivere codice per Arduino utilizzando la programmazione a blocchi in stile Scratch.

6. Programmare Arduino a blocchi con Bitbloq

Bitbloq è un ambiente di programmazione online, a blocchi, per Arduino. Si può utilizzare accedendo all'indirizzo <http://bitbloq.bq.com> con il browser Chrome.

In dettaglio

Bitbloq 2 è un linguaggio di programmazione a blocchi che si può utilizzare per comporre degli sketch per Arduino. Si utilizza online e può programmare direttamente le schede Arduino Uno tramite un plugin da installare sul proprio computer. La forma dei blocchi è molto simile a quella di Scratch e di TinkerCAD Circuits. A differenza di TinkerCAD i circuiti e gli sketch non possono essere simulati ma solo composti. Per la verifica è necessario caricarli su una scheda reale che può essere un Arduino Uno, un Arduino Mega o alcuni kit robotici proposti da Bitbloq. Le schede e l'hardware sono limitati e si possono usare solo i componenti base presenti nei kit. Non troverete breadboard, resistenze e transistor. Per programmare con Bitbloq è necessario installare il browser Chrome ed è consigliabile registrarsi, benché si possa utilizzare anche come "guest" (ospite).

L'area di lavoro è divisa in tre colonne principali:

- sulla sinistra trovate un selettore per attivare le viste "hardware", "software" e "project info";
- al centro, l'area principale, in cui comporre circuiti e blocchi di codice;

- a destra un pannello con le parti hardware o i vari tipi di blocchi (a seconda della modalità di lavoro attiva).

Per programmare Arduino con Bitbloq:

1. collegate una scheda Arduino Uno al computer con il cavo USB;
2. aprite Bitbloq 2 (<http://bitbloq.bq.com>) utilizzando il browser Chrome;
3. accedete inserendo login e password, oppure lavorate come "ospite";
4. se vi siete "loggati", create una nuovo progetto;
5. cliccate sulla linguetta "Boards" e trascinate una scheda Arduino Uno nell'area di lavoro (Figura 1.15);



Figura 1.15 - L'area principale di lavoro di Bitbloq; trascinate una scheda Arduino Uno per iniziare a comporre codice.

6. aggiungete i componenti hardware necessari, trascinandoli nell'area di lavoro e collegandoli tra di loro;
7. passate alla vista "software" e componete uno sketch trascinando i blocchi dalla colonna laterale;
8. cliccate su "Verifica" (pulsante a forma di "V" posto sul menu/toolbar superiore) e attendete che il programma sia compilato correttamente;
9. cliccate sul pulsante "Carica" (ha un simbolo a forma di freccia ed è posto sul menu/toolbar superiore) e apparirà una finestra che chiederà di scaricare l'applicazione Web2Board. Selezionate il sistema operativo corretto e installate il plugin. Potrebbe essere necessario riavviare il computer. Questa operazione è necessaria solo la prima volta che si utilizza Bitbloq;

10. dopo il riavvio riaprite Bitbloq 2, riaprite il progetto e cliccate di nuovo sul pulsante "Carica";
11. al termine dell'upload, apparirà un messaggio sullo schermo dicendo che tutto è andato a buon fine: "Programma caricato con successo".

Noterete che i collegamenti elettrici sono semplificati. Per collegare un LED ad Arduino viene indicato un solo filo, questo perché è previsto l'utilizzo con delle schede particolari e kit robotici che non prevedono un assemblaggio "complesso" dei circuiti.

La sezione di programmazione a blocchi è divisa in tre parti: una zona per definire variabili globali, funzioni e classi con visibilità globale, una zona **Setup** e una zona **Loop**. I blocchi sono raccolti nella colonna di destra e divisi per funzionalità. Per aggiungere un blocco allo sketch è sufficiente trascinarlo in una delle tre zone di lavoro, esattamente sul rettangolo "Drag a block here to start your program".

I blocchi inseriti nella zona **Setup** sono eseguiti all'inizio del programma e sono utilizzati per determinare il comportamento dei pin e di altre azioni che devono essere eseguite una sola volta all'inizio del programma. I blocchi inseriti nella zona **Loop** sono ripetuti continuamente.

Se siete curiosi di vedere come viene composto lo sketch, potete passare dalla vista blocchi a quella codice cliccando su "Blocks" o "Code".

Per far lampeggiare un LED dovreste prima di tutto collegare un LED alla scheda (Figura 1.16) e poi passare alla sezione Software per comporre i blocchi necessari (Figura 1.17).



Figura 1.16 – Aggiungete componenti elettronici trascinandoli dal pannello laterale.

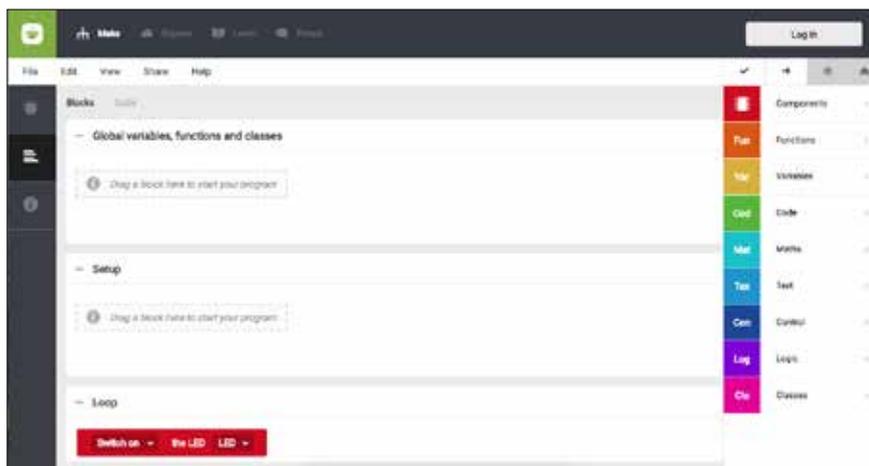


Figura 1.17 – Nella sezione software potete comporre uno sketch trascinando blocchi dal pannello laterale.

Non è necessario dichiarare il pin utilizzato nella sezione **Setup**. Per realizzare il “blink” trascinate un blocco di “Switch on” dall’area “Components” (di colore rosso), quindi un blocco “Delay” dalla sezione “Control” (colore blu), poi ancora un blocco “switch off” e un altro “Delay” (Figura 1.18). Contemporaneamente allo sketch grafico, Bitbloq genera il corrispondente codice, visibile nella sezione “Code” (Figura 1.19).

Completate il progetto aggiungendo nome e descrizione nella sezione “Project Info”.

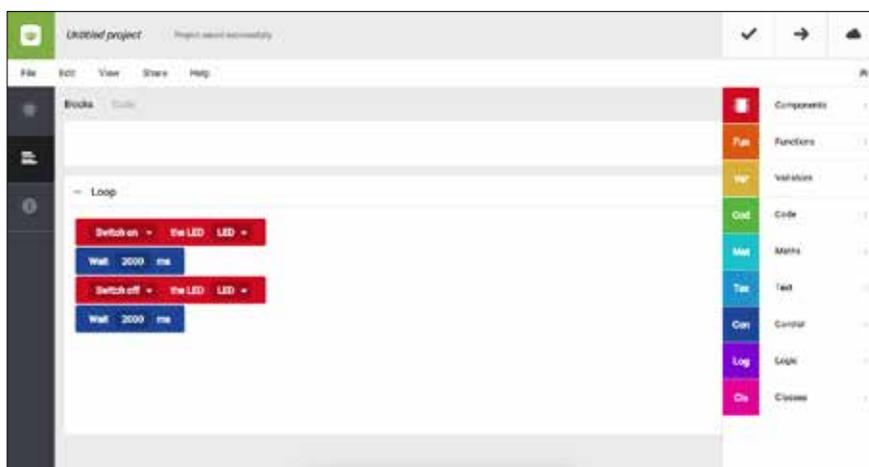
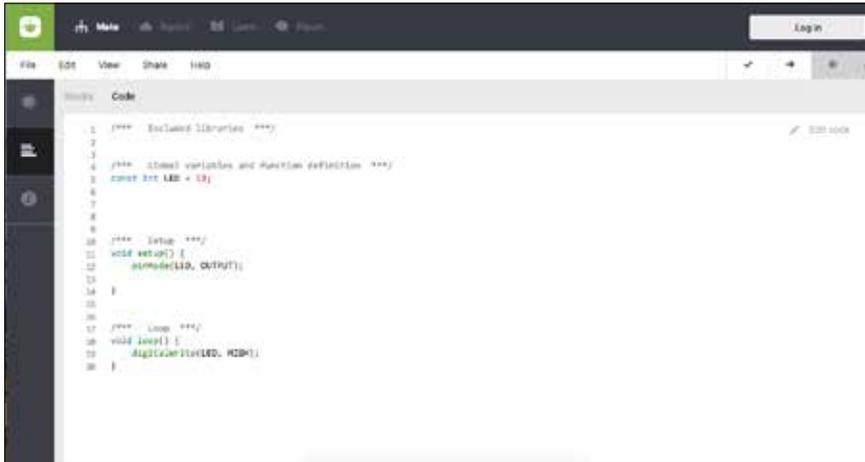


Figura 1.18 – Il codice di un semplice “blink”.



The image shows a screenshot of a code editor interface. The editor has a dark theme and a sidebar on the left with icons for file explorer, search, and other functions. The main area displays C++ code for an Arduino project. The code includes comments in Italian and standard Arduino functions like `setup()` and `loop()`. The code is as follows:

```
1 /** Excluded Libraries **/  
2  
3  
4 /** (Some) variables and function definition **/  
5 const int LED = 13;  
6  
7  
8  
9  
10  
11 /** Setup **/  
12 void setup() {  
13   pinMode(LED, OUTPUT);  
14 }  
15  
16  
17 /** Loop **/  
18 void loop() {  
19   digitalWrite(LED, HIGH);  
20 }
```

Figura 1.19 – Se siete curiosi, potete dare una sbirciata al codice generato da Bitbloq.