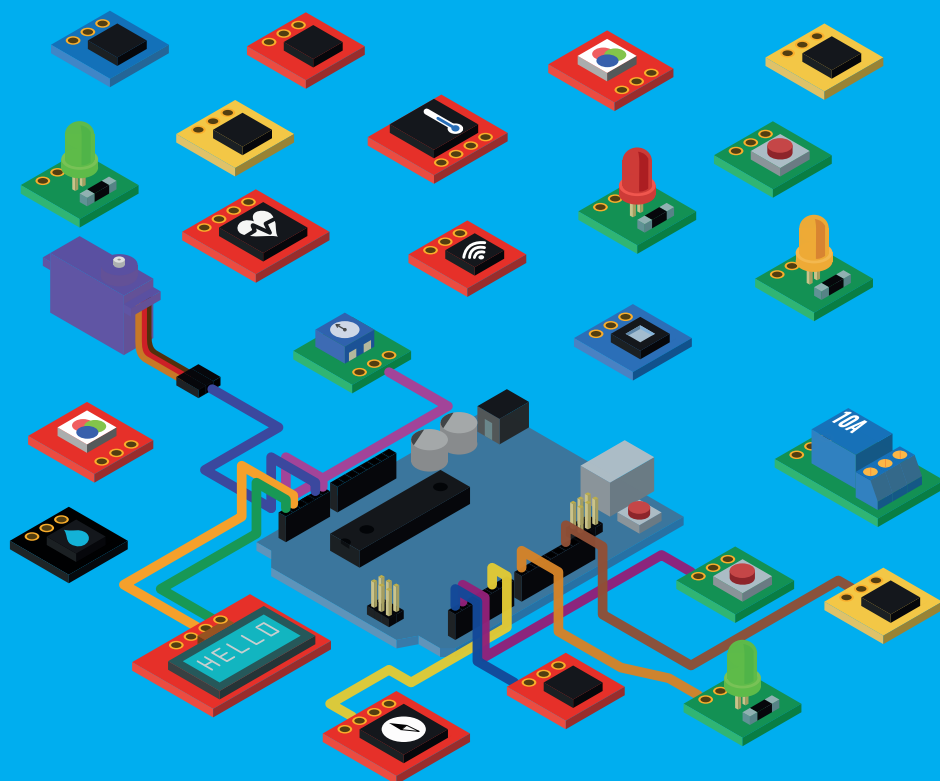


Paolo Aliverti

Progettare con Arduino

LED, display, sensori: 30 progetti completi



Dall'effetto Supercar al sonar, fino al cardiofrequenzimetro >>

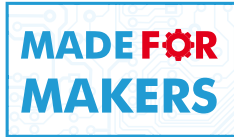
Realizzazioni con sensori, LED, moduli e display >>

Con illustrazioni a colori e schemi elettronici >>

Ogni progetto è corredato da un video >>

MADE FOR
MAKERS

EDIZIONI
LSWR



Progettare con Arduino

LED, display, sensori: 30 progetti completi

Paolo Aliverti

EDIZIONI
LSWR

Progettare con Arduino | LED, display, sensori: 30 progetti completi

Autore: Paolo Aliverti

Collana:



Publisher: Marco Aleotti

Progetto grafico: Roberta Venturieri

Immagine di copertina: Paolo Aliverti

Logo design: Giampiero Carella

© 2021 Edizioni LSWR* - Tutti i diritti riservati

ISBN: 978-88-6895-916-6

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 febbraio 2021 presso "LegoDigit" Srl., Lavis (TN)

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

Indice

1. PROGETTI CHE NON FUNZIONANO	7
Il canale video	9
Per chi è pensato questo libro?	10
Il materiale necessario	11
Affrontare i progetti	38
Chi sono io?	40
2. GIOCHI E CIRCUITI CON I LED	41
1. Candela elettronica	42
2. Dissolvenza multicolore con LED RGB	46
3. Dado elettronico a LED	51
4. Dado elettronico con registro a scorrimento	59
5. Dado elettronico con matrice di LED e accelerometro	67
6. Clessidra a matrice di LED	81
7. Sistema semaforico a tre tempi	104
8. Scritte scorrevoli su display a matrice	111
9. Effetto "SuperCar" con LED NeoPixel	134
10. Effetto fiamma a LED e con NeoPixel	144
11. Orologio a NeoPixel (un disco per le ore, uno per i minuti e uno per i secondi)	149
12. Caleidoscopio elettronico	159
3. VISUALIZZARE INFORMAZIONI	165
13. Menu a livelli con display LCD e KeyPad Shield	166
14. Generatore di password con display LCD	186
15. Orologio impostabile con data e sveglia	197
16. Visualizzare dati su un grafico cartesiano con un display OLED	215
17. SONAR grafico, ultrasonico	221
18. Combinatore di cifre a singolo pulsante	232
19. Pong con display OLED	241
20. Invasori Spaziali	249
21. Arduino come joystick per Processing	259
22. Calcolatrice con display a sfioramento	267

4. SENSORI.....	281
23. Clap control.....	283
24. Sensore di parcheggio (ultrasuoni)	289
25. Misuratore di pH con display LCD.....	298
26. Termometro con display a sette segmenti	306
27. Controller grafico con display a sfioramento.....	314
28. Riconoscere i colori e riprodurli con un LED RGB	320
29. Cardiosfrequenzimetro.....	326
30. Apertura porta con impronta digitale.....	337
31. Bussola a NeoPixel.....	344
32. Ohmetro	352
APPENDICE	359
Come trovare i video?	359
INDICE ANALITICO	363

Progetti che non funzionano

Negli anni '80 del secolo scorso (e mentre lo scrivo mi sento già vecchio), le edicole erano il mio punto di spaccio culturale preferito. Avevo varie passioni e ogni mese non sapevo mai come spendere la mia paghetta... su quale rivista buttarmi? Ai quei tempi frequentavo le scuole medie e ogni mattina prendevo un autobus per andare da Ceriano Laghetto a Saronno. Le lezioni iniziavano alle 8 e certi inverni pareva di essere in Siberia: -10, -12 gradi, vento polare, buio pesto... Il capolinea dell'autobus era presso la stazione ferroviaria dove, prima di incamminarmi per la scuola, mi fermavo nell'atrio della biglietteria per curiosare in edicola e scegliere che rivista comprare. La scelta era sempre preceduta da alcuni giorni di meditazione e tremenda indecisione... il budget era limitato! Conservo ancora molte riviste e ho passato ore e ore a leggerle e rileggerle per cercare di capire cosa raccontavano. Le mie preferite erano quelle di elettronica e di informatica. Purtroppo in entrambi i casi le letture degli articoli e le descrizioni dei progetti erano spesso aride e non ci capivo nulla. Cercavo di comprendere perché in un certo progetto fossero state fatte alcune scelte o perché funzionasse in quel modo, ma raramente le spiegazioni scendevano in dettaglio, forse dando per scontate molte cose che ancora non sapevo: avevo undici anni! Ancora peggio erano le raccolte di listati per i computer in voga in quegli anni (Commodore 64 e ZX Spectrum): centinaia di linee di codice da ricopiare senza nessuna spiegazione! Il codice, una volta copiato, per errori di battitura o tipografici, aveva una probabilità quasi nulla di funzionare.

Mi accanivo cercando di capire come questi circuiti funzionassero e forse quell'accanimento ha un po' segnato la mia vita e tutte le scelte che ho fatto in seguito. Credo che dover affrontare quegli enigmi avendo a disposizione pochissime fonti da cui trarre spiegazioni mi abbia aiutato a sviluppare una pazienza tendente all'infinito, tipica di ricercatori, esploratori, alpinisti e di chi vuole capire e non si stancherà fino a che non "arriverà in cima". Oggi è tutto più facile e questo è un bene perché possiamo avere accesso a tutta la conoscenza del mondo in un istante grazie a Internet, ma abbiamo perso la pazienza e la forza di ragionare in autonomia sulle cose. Se non c'è su Internet quello che cerchiamo... allora non esiste.

Ho notato che in Rete si trovano moltissimi progetti elettronici con Arduino ma non sempre sono documentati e permettono di capire. Ci sono migliaia di video su YouTube che mostrano fantastiche realizzazioni senza spendere una parola su come si possa fare per replicarle. Credo che certi contenuti siano stati creati solo per raccogliere visualizzazioni, senza uno scopo didattico. Questo ci riporta un po' alle riviste degli anni '80... ed è per questo che è nato questo libro: per offrirvi progetti descritti nei minimi particolari, illustrando come sono nati e qual è il ragionamento che sta dietro al codice. Spero possiate capire come funzionano per poi modificarli e farli vostri. Troverete che i miei non sono progetti perfetti, sono completi all'80%, si potrebbero migliorare... ma questo è il mio intento e il vostro compito!

Ho cercato di dividere i progetti per argomenti generali. Dopo il titolo di ogni progetto ho inserito una breve descrizione del suo funzionamento e una classificazione della difficoltà complessiva. Troverete tre livelli che valutano nel complesso la parte elettronica e la stesura del firmware:

- ■ : progetto nel complesso semplice, senza particolari difficoltà nell'assemblaggio e nella struttura dello sketch;
- ■ ■ : progetto che richiede un certo impegno nella realizzazione elettronica e nella scrittura dello sketch;
- ■ ■ ■ : progetto complesso perché richiede attenzione nella realizzazione elettronica, anche per le complessità circuitali, o perché prevede uno sketch esteso o con difficoltà teoriche.

Se non siete molto esperti vi consiglio di iniziare ad affrontare i progetti con difficoltà crescente. Per apprendere meglio provate, una volta che li avete compresi, a modificarli, creare variazioni o aggiungere funzionalità. Alcune volte troverete spunti per proseguire, in fondo al progetto, subito dopo al listato completo. Per ogni progetto ho cercato di presentare il ragionamento seguito nella scrittura del firmware che secondo me è la parte più critica.

Anche questo nuovo lavoro nasce dai video pubblicati sul mio canale YouTube (www.youtube.com/user/zeppelinmaker). Ho notato che questo approccio mi mette al riparo da spiacevoli errori (almeno dalla maggior parte di essi) e offre un prodotto di migliore qualità. Gli errori ci saranno sempre, ma con questo metodo credo di poterli ridurre notevolmente, relegandoli a refusi o piccole disattenzioni.

In fondo al libro troverete una tabella che riporta per ogni progetto i video di riferimento, così che possiate avere a disposizione anche questo tipo di risorsa per poter capire meglio e poter osservare il risultato finale. Nella descrizione di ogni video troverete un riferimento al mio repository di GitHub con tutti i file sorgenti e gli schemi circuitali. Vi raccomando caldamente di provare a scrivere voi il codice e di non limitarvi a fare

del copia e incolla senza capire o, peggio ancora, a leggere passivamente (o osservare i video senza provarli).

Se non siete esperti e non avete ancora acquistato *Arduino trucchi e segreti* (Edizioni LSWR, 2018), ve lo consiglio caldamente perché può essere di grande aiuto per comprendere i progetti. Non sono certo la persona più adatta per consigliarvelo, avendolo scritto io, ma se sbirciate le recensioni su Amazon scoprirete che è un testo molto apprezzato e in fondo ben fatto (e questo anche grazie ai numerosi commenti che ogni giorno mi pervengono tramite YouTube e i social media e che mi aiutano a migliorare).

Il canale video

I progetti di questo libro sono stati tutti presentati come video tutorial sul mio canale YouTube (www.youtube.com/user/zeppelinmaker) dove ho iniziato a pubblicare materiale a partire dal 2011, prima abbastanza casualmente, poi con sempre maggior frequenza e regolarità. I miei primi video erano abbastanza insensati: filmavo quello che avevo costruito senza alcuna spiegazione o commento. Negli anni YouTube è cresciuto ed è diventato un vero e proprio "social medium". Mi sono accorto che poteva essere un canale di comunicazione potente e molto diretto e quindi ho studiato i video degli YouTuber di successo per capire come impostare i miei.

Per girare un video non ci si può improvvisare. Anche se molti sembrano molto case-recci, dietro a un video di successo c'è sempre una certa organizzazione. Ho cercato di creare qualcosa di valido con un minimo di studio e organizzazione. Una semplice macchina fotografica non basta. È fondamentale ma deve essere accompagnata anche da un buon audio. Ho recuperato un ottimo microfono e vari cavalletti. Un'altra cosa importante è l'illuminazione a cui ho provveduto con varie lampade e vi assicuro che sono sempre poche!

I miei primi video erano girati quasi al buio, più per necessità che per scelta. L'unico posto a mia disposizione era un locale nel mio sottotetto e potevo girare video solo quando tutti dormivano. Riguardandoli oggi vedo quanto siano cupi e anche un po' inquietanti. La mia voce era molto dimessa perché non potevo parlare con un volume adeguato! Un altro elemento fondamentale è il montaggio, che richiede tempo e pazienza. Audio e video devono essere uniti e il tutto va ripulito da rumori, balbettii, pause ed errori. Non c'è un vero e proprio copione ma, per evitare di divagare, ogni video è preparato prima su carta e provato prima della registrazione.

Mi sono attrezzato così perché volevo creare qualcosa di "buono", semplice e con una certa qualità. La maggior parte dei video "tecnici" girati in Italia sono molto approssimativi e abbastanza deludenti. Spesso non si giunge al punto, ci sono lunghe divagazioni. Alcuni di questi sono girati con un cellulare senza cavalletto: le immagini traballano e

oscillano. Vedendo questi video mi sono detto: “Di video capisco poco, non sono bello come Johnny Depp... ma posso fare qualcosa di meglio della media”. Ho iniziato a creare tanti video: uno al giorno, tutti i giorni, pubblicati alle 9 in punto. Viste le premesse, un impegno notevole!

Questa tattica ha fatto crescere il mio canale fino a 40.000 iscritti (dicembre 2020) e più di 6.600.000 visualizzazioni (sono sempre una goccia nell’oceano di YouTube). Una piccola platea di appassionati mi scrive, commenta e critica ogni giorno. I messaggi che ricevo quotidianamente sono la miglior ricompensa per tutto il tempo che investo con passione nella mia attività divulgativa.

La mia attrezzatura

Spesso mi chiedono che attrezzatura utilizzo. Non ho segreti:

- fotocamera Canon D100;
- microfono Røde Lavalier;
- mini mixer Xenyx con interfaccia audio USB;
- OpenBoard come lavagna con tavoletta grafica Wacom Intuos;
- QuickTime per la registrazione dello schermo;
- iMovie per il montaggio dei video.

Per chi è pensato questo libro?

Questo libro nasce come prosieguo naturale dei miei due precedenti volumi, *Il manuale di Arduino* e *Arduino Trucchi e segreti*. Il primo dei due, *Il manuale di Arduino*, è una guida introduttiva per avvicinarsi e conoscere lo strumento. L’ho scritto seguendo la traccia dei corsi che tenevo e che ancora presento. Il partecipante tipico è a digiuno di Arduino e della programmazione e vuole iniziare a capire come funziona questo strumento. Per questo motivo il libro è molto discorsivo e presenta gli elementi fondamentali necessari per programmare il microcontrollore e per costruire i primi semplici circuiti. Il secondo volume è nato come approfondimento del primo. Nonostante nel primo siano state inserite parecchie pagine dedicate all’uso di sensori e attuatori, le possibilità con Arduino sono pressoché infinite e quindi ho esplorato maggiormente questo aspetto, aggiungendo nuove informazioni. *Arduino Trucchi e segreti* è pensato per chi ha già iniziato a utilizzare Arduino e vuole approfondire e migliorare le sue conoscenze. Nonostante questa assunzione iniziale ho cercato di presentare gli argomenti in modo semplice e comprensibile anche a chi magari sta muovendo i primi passi e se lo trova tra le mani. Quest’ultimo lavoro invece presuppone una minima conoscenza dell’utilizzo di Arduino. Non è necessario essere esperti, ma non mi soffermerò di nuovo su come si carica uno sketch, si installa una libreria o su come funziona un pulsante in pull-up.

Se necessario troverete brevi indicazioni su come procedere. L'attenzione è andata sul progetto, cioè sulla realizzazione di oggetti più complessi che richiedono sensori, attuatori, dispositivi e lo sviluppo di sketch molto articolati.

I progetti nascono dalle richieste prevenute sul mio canale YouTube oppure sono soluzioni a casi pratici o reali. Ho riscontrato un notevole interesse per l'argomento "progetti", perché molti desiderano capire come funzionano certi meccanismi e replicarli. Ho cercato comunque di dettagliare ogni lavoro illustrando l'idea iniziale e raccontando lo sviluppo, sperando che questo modo di procedere, molto discorsivo, possa aiutare il lettore a capire come ragionare e come affrontare questo tipo di problemi.

Non pretendo di essere un guru della programmazione o dell'elettronica e so che molti potranno obiettare che esistono modi migliori per sviluppare alcuni degli argomenti incontrati: ci sono magari soluzioni più compatte, più rapide o più eleganti. Ovviamente non c'è mai un solo modo per sviluppare le cose e ognuno segue la strada che preferisce. Oggi, dovendo tracciare io la via, ho cercato di seguire quella meno impervia, quella più sicura, con la minor pendenza, sperando di portare tutti in vetta.

Il materiale necessario

Per sviluppare i progetti raccolti in questo libro è necessario disporre di una dotazione minima di hardware elettronico. Nella maggior parte dei casi saranno sufficienti pochi e semplici componenti che potreste avere già nelle vostre cassettiere: resistori, pulsanti, potenziometri e qualche LED. L'elemento fondamentale è però la scheda Arduino che sarà sempre al centro di ogni realizzazione. Per affrontare i progetti presenti nel libro, una scheda Arduino UNO o Arduino Nano è più che sufficiente.

La piattaforma Arduino

La scheda Arduino viene creata nel 2005, a Ivrea, presso l'Interaction Design Institute, con l'intento di creare un oggetto elettronico programmabile con semplicità. Il modello più noto e diffuso si chiama "Arduino UNO" ed è oggi quasi uno standard per realizzare prototipi rapidi nel modo dei maker e degli hobbisti. Anche molti professionisti si avvicinano a questo "sistema" per la sua semplicità di utilizzo. Il team di Ivrea in realtà non ha inventato nulla di straordinario ma ha saputo unire gli elementi giusti nel momento opportuno, smussando alcuni aspetti critici e alcuni pregiudizi esistenti sull'utilizzo dell'elettronica nei progetti interattivi. Il cuore di Arduino è il microcontrollore ATmega328, prodotto un tempo da Atmel, ora acquisita da Microchip. La scheda Arduino UNO ospita questo chip e gli fornisce un'adeguata alimentazione, completandolo con una porta seriale accessibile via USB. L'oggetto, semplicissimo, può essere programmato direttamente tramite un cavo USB e può svolgere innumerevoli compiti.

Con questo non intendo sminuire il lavoro compiuto che in realtà è notevole e ammirevole. Solitamente la programmazione dei microcontrollori è demandata a ingegneri elettronici o informatici. Servono programmi dedicati e molto complessi, oltre ai “programmatori”, cioè circuiti che si occupano di caricare il software dal computer utilizzato per lo sviluppo “a bordo” del microcontrollore. Grazie all’introduzione di un piccolo software pre-installato a bordo dell’ATmega328 (il bootloader), possiamo trasferire rapidamente il firmware senza necessità di acquistare un hardware di programmazione dedicato. Per la scrittura del software è stato adottato l’ambiente Processing, adattandolo e modificandolo quanto necessario per utilizzarlo in combinazione con Arduino. Processing è un ambiente didattico nato per apprendere la programmazione in Java molto facile da utilizzare. I programmi di Processing si chiamano “sketch” e hanno una struttura semplice che prevede una sezione per l’inizializzazione e una sezione continuamente ripetuta chiamata `draw()` (in Arduino poi rinominata `loop()`). In origine Arduino ha utilizzato Wiring, un lavoro realizzato nel 2003 da Hernando Barragán sempre presso l’Interaction Design Institute di Ivrea. Lo sviluppo di Barragán include una libreria che semplifica la scrittura del codice per i progetti destinati a un microcontrollore. Con questo metodo non si avrà a che fare con byte, bit, nomi criptici e sintassi incomprensibili tipici di questi ambienti di programmazione, ma con espressioni quasi “parlanti” per esprimere in modo immediato e diretto quello che si vuole realizzare con il microcontrollore. Arduino, utilizzando questo sistema semplificato, è risultato più attraente e digeribile da un insieme di persone interessate allo sviluppo di progetti interattivi che, fino al 2005, non si sarebbero mai azzardate ad addentrarsi nelle complessità necessarie per programmare i microcontrollori. L’ambiente di sviluppo per Arduino è denominato “Arduino IDE” ed è possibile scaricarlo gratuitamente dal sito ufficiale (www.arduino.cc/en/software). Questa versione del programma si può installare sul computer e non richiede una connessione Internet per funzionare. Offre funzionalità di base per scrivere il codice, evidenziando le parole chiave del linguaggio e facilitando l’importazione delle librerie tramite un sistema denominato “Library Manager”. Attualmente è in fase di completamento una nuova versione del software di sviluppo, chiamata “Arduino IDE PRO”. Il nuovo ambiente di programmazione offrirebbe la possibilità di debuggare il codice passo per passo direttamente sull’hardware e una migliore organizzazione delle informazioni. Esiste anche la possibilità di utilizzare un editor on line, che replica le funzioni dell’Arduino IDE. Per farlo funzionare però è necessario installare un “agente”, cioè un servizio in esecuzione sul computer che trasferirà il firmware prodotto on line sull’Arduino collegato alla macchina. Per farlo utilizzare questo sistema è necessario avere una buona connessione Internet. Il vantaggio sta nella possibilità di salvare in cloud i propri progetti e di averli sempre

a disposizione. La compilazione da un lato risulta facilitata perché avviene sui server di Arduino e non dobbiamo installare diversi sistemi a seconda della scheda utilizzata, dall'altro è molto più lenta di quella effettuata direttamente sul proprio pc.

Le istruzioni fondamentali di Arduino permettono di controllare i pin. Queste istruzioni sono quelle maggiormente utilizzate in un listato (chiamato "sketch"), in combinazione con le parole chiave del linguaggio C/C++. Abbiamo:

- `pinMode()`: per impostare il comportamento di un pin;
- `digitalWrite()`: per attivare un'uscita;
- `digitalRead()`: per leggere un segnale digitale in arrivo su un pin;
- `analogWrite()`: per impostare un segnale particolare (PWM) su alcune uscite, utile per controllare LED e motori;
- `analogRead()`: per convertire una tensione analogica in un numero utilizzabile all'interno del codice.

Queste istruzioni, unite a numerosissime librerie e hardware di ogni tipo, hanno reso possibile sviluppare molteplici progetti. Il codice disponibile in Rete, assieme a numerosissimi esempi, è sempre distribuito in un'ottica open source, dove il "copia e incolla" e il rimescolamento di frammenti di listati presi da svariate fonti è addirittura incoraggiato. Dopo questa panoramica iniziale, vediamo quale Arduino potrebbe essere adatto allo sviluppo dei progetti presentati. In tutti i casi ho sempre utilizzato un Arduino UNO o un Arduino Nano. Queste due schede sono del tutto equivalenti come prestazioni. Esistono sul mercato numerosissimi cloni delle schede originali. Alcuni di questi cloni nel tempo si sono anche costruiti una buona reputazione, offrendo alla fine alternative economiche per chi vuole iniziare. Quali sono le schede principali? Quale conviene utilizzare?

Arduino UNO

La scheda Arduino UNO è la più nota e popolare. Il suo colore verde-blu e la caratteristica forma smussata la rendono un'icona del mondo maker. Il microcontrollore utilizzato nella versione più comune è l'ATmega328P, un chip con venti piedini, inserito tramite uno zoccolino, ma esistono anche versioni con chip a montaggio superficiale (SMD). La scheda lavora a 16 MHz e il processore ha un'architettura a 8 bit. La memoria RAM è di 2 kB mentre ci sono 32 kB di flash memory (usata per immagazzinare i programmi caricati). Arduino UNO ha 14 pin digitali, di cui 6 utilizzabili per generare segnali PWM e 6 ingressi analogici che possono essere utilizzati anche come I/O digitali. Per comunicare con il mondo esterno o altre periferiche possiamo utilizzare un bus I²C, un SPI e una porta seriale. La tensione di alimentazione è prelevata dalla porta USB o da un

alimentatore esterno ed è di 5 V. Tutto questo sembra poco ma queste caratteristiche sono più che sufficienti per realizzare praticamente quasi ogni tipo di progetto.

Una variante molto nota è la scheda Arduino Leonardo che utilizza il microcontrollore ATmega32u4, sostanzialmente comparabile con l'ATmega328 ma con 0,5 kB di memoria in più. La scheda Leonardo può essere programmata in modo che sia riconosciuta come un mouse o una tastiera quando è collegata a un computer.

Per la realizzazione di progetti con le schede Arduino UNO è necessario utilizzare una breadboard da affiancare alla scheda. Le connessioni sono in genere realizzate utilizzando cavetti (jumper) intestati con piccole spinette metalliche. I prototipi possono quindi risultare più delicati e soggetti a disturbi.

Tabella 1.1 - Caratteristiche della scheda Arduino UNO.

Microcontrollore	ATmega328P
Frequenza di clock	16 MHz
Tensione di funzionamento	5 V
Memoria RAM	2 kB
Memoria flash (programma)	32 kB
Memoria EEPROM (permanente)	1 kB
Numero pin digitali	14+6
Ingressi analogici	6
Pin PWM	6



Figura 1.1 - La scheda Arduino UNO Rev3.