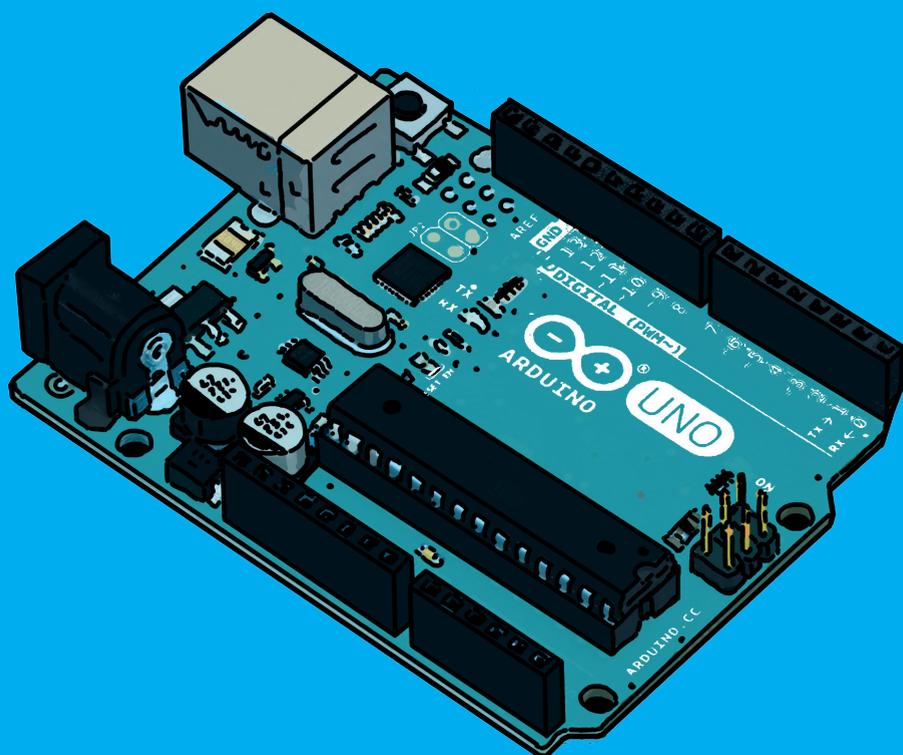


Paolo Aliverti

Il manuale di Arduino



Programmazione di base e avanzata con C e C++ >>

Sensori, attuatori, trasmettitori e Internet >>

Collegamento di shield e breakout board >>

Arduino Yún e Internet degli oggetti >>

**MADE FOR
MAKERS**

EDIZIONI
LSWR



Il manuale di Arduino

Paolo Aliverti

EDIZIONI
LSWR

Il manuale di Arduino

Autore: Paolo Aliverti

Collana:



Editor in Chief: Marco Aleotti

Progetto grafico: Roberta Venturieri

Immagine di copertina: Paolo Aliverti

Logo design: Giampiero Carella

Realizzazione editoriale e impaginazione: Studio Dedita di Davide Gianetti

© Edizioni Lswr* - Tutti i diritti riservati

ISBN: 978-88-6895-340-9

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.

EDIZIONI
LSWR

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

Printed in Italy

Finito di stampare nel mese di maggio 2016 presso "LegoDigit" Srl., Lavis (TN) - Italy

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

Sommario

INTRODUZIONE	9
1. IL MONDO DI ARDUINO	13
La storia	15
Microcontrollori e microprocessori	17
Shield	18
Maker & Arduino	19
Siti di riferimento	21
2. HARDWARE.....	23
Arduino Uno (o Genuino Uno)	24
Una scheda non basta!	27
Arduino è Atmel	40
L'elettronica non è il mio forte: shield, TinkerKit e Grove	41
3. INSTALLAZIONE.....	47
Download e installazione	48
Hello LED!.....	52
4. ARDUINO E IL LINGUAGGIO C/C++	59
Imparare a programmare	61
Il linguaggio C/C++	66
5. PROGRAMMARE ARDUINO.....	87
Lo sketch.....	87
pinMode	91
delay	94
digitalRead.....	101
analogWrite.....	103
Serial Monitor	106
analogRead.....	109
Prendere il tempo	111
Misurare la lunghezza di un impulso.....	117

Funzioni matematiche.....	118
Funzioni per i caratteri	120
Funzioni per manipolare bit e byte.....	121
Conversioni di tipo.....	124
sizeof	125
6. OLTRE LE BASI.....	127
Collegare un LED.....	127
LED ondeggiante	129
LED ondeggiante - versione software.....	131
LED RGB.....	133
Pulsanti	134
Pulsante senza resistenza	136
Pulsante su linea analogica.....	139
Utilizzare un servomotore.....	140
Display	143
LCD	148
Registri a scorrimento (shift register).....	152
Moduli.....	156
Librerie	158
7. PROGRAMMAZIONE AVANZATA.....	163
Interrupt.....	163
EEPROM e variabili persistenti	166
Assembler	169
Scheduler.....	171
Timer	172
analogReference.....	189
Il bootloader.....	190
Programmare i chip AVR.....	192
8. SENSORI.....	201
Trimmer e potenziometri.....	201
Encoder	203
Luce.....	204
Temperatura	207
Umidità	209
Gas.....	211
Sensore di Hall	212
Tilt sensor.....	213
PIR	213
Sensori capacitivi	214
Misuratore di distanza a infrarossi	216
Ultrasuoni.....	219
Accelerometro e giroscopio	221

Bussola.....	222
GPS.....	223
Tempo e data.....	224
Collegare un microfono o una fonte audio.....	227
RFID	228
Schede di memoria SD.....	230
Altri sensori	233
9. ATTUATORI.....	235
Il transistor bipolare.....	235
Pilotare un relè	238
I transistor MOSFET	241
Pilotare un motore passo-passo.....	246
Uscite optoisolate	247
Potenzimetri digitali	248
Collegare un DAC.....	250
Uscita audio	251
Altri attuatori.....	252
10. COMUNICARE	253
Un po' di teoria.....	253
Porte seriali	255
Software Serial	259
Convertitori di livelli.....	260
I ² C.....	263
SPI	267
RS232 & Co.....	269
Bluetooth	269
ZigBee	274
Ethernet	274
WiFi	281
GSM/GPRS	282
Prossimi passi.....	284
11. ARDUINO YÚN	285
Arduino piú Linux	285
Installazione.....	286
Programmazione WiFi	288
OpenWrt-Yun	288
Console	295
Semplice web server.....	296
Pagine web dinamiche	298
Reset	301
Bridge	302
Pronti per partire?	307

CONCLUSIONE.....	309
APPENDICE: LEGGERE IL VALORE DI UNA RESISTENZA	311
BIBLIOGRAFIA CONSIGLIATA	315
GLOSSARIO	319
INDICE ANALITICO	323

Introduzione

Questo libro è rimasto nel cassetto per lungo tempo, forse anche troppo. È nato il primo giorno in cui ho iniziato a tenere corsi su Arduino, nel 2011, e raccoglie tutte le domande, i dubbi e le difficoltà incontrate per cercare di spiegare nel modo più semplice possibile argomenti abbastanza complessi a un pubblico non tecnico. I partecipanti dei miei corsi, infatti, non erano ingegneri, ma designer, inventori, architetti, artisti, creativi... Persone che raramente nel loro percorso formativo (istituzionale oppure no) avevano incontrato l'elettronica. Grazie ai loro dubbi, alle loro domande e alla loro visione del mondo molto più aperta della mia, da ingegnere, ho imparato molte cose e soprattutto a guardare in un'ottica differente quello che sapevo.

Molti spunti e contenuti sono nati anche dai progetti che ho realizzato in questi anni, a partire dal 2009, quando ho acquistato il mio primo Arduino e sono rimasto sorpreso dalla semplicità di questa soluzione. Ho sempre "trafficato" con l'elettronica e sapevo che programmare microcontrollori non era una cosa molto semplice, per via delle attrezzature e delle competenze necessarie.

Arduino ha cambiato molte cose: si può programmare nel giro di pochi secondi con un semplice cavo USB. In più, le istruzioni necessarie sono abbastanza lineari e comprensibili, tanto da sembrare quasi linguaggio JavaScript invece che C. Rivoluzionario. Infatti, molti maker lo hanno eletto a "coltellino svizzero" della loro cassetta di attrezzi e lo impiegano in numerosi progetti, dai più semplici per far lampeggiare luci e LED fino a schede di controllo per stampanti e droni.

Il manuale di Arduino è un po' la prosecuzione del mio precedente lavoro *Elettronica per maker*, uscito l'anno scorso, con lo scopo di preparare il terreno agli aspiranti maker. Molti dei miei "studenti", infatti, avevano bisogno di capire i concetti base dell'elettronica per poter arrivare ad Arduino. I due libri costituiscono quindi una specie di "super-manuale" di Arduino, che fornisce al lettore tutto quello che serve per creare un progetto elettronico.

Questo manuale è un testo intermedio, anche se ho cercato di rendere tutto il più semplice possibile e di non lasciare nulla per scontato. A volte queste semplificazioni

potrebbero essere snervanti per chi è più esperto, ma ho cercato di pensare sempre ai lettori “meno tecnici”. Ho voluto tenere un approccio operativo, per mettervi nelle condizioni di capire e quindi fare. Nel libro *Elettronica per maker* ho commesso qualche errore, soprattutto nei disegni, tutti realizzati da me. Ogni tanto qualche errore sfugge a causa della complessità del lavoro di scrittura di un intero libro. Anche in questo libro ci saranno degli errori e mi scuso in anticipo: nel caso ne troviate qualcuno, non esitate a segnalarmelo sul mio sito www.zepelinmaker.it o alla mail paolo@zepelinmaker.it. Anche qui ho cercato di disegnare tutte le immagini necessarie. Qualche volta, però, ho utilizzato Fritzing perché, oltre a essere molto utile per “tracciare” prototipi su breadboard, ha delle immagini vettoriali molto belle.

Argomenti del libro e descrizione dei capitoli

Il Capitolo 1 racconta la storia di Arduino, cos'è un microcontrollore e per cosa può essere utilizzato. Nel Capitolo 2 vedrete da vicino l'hardware di Arduino Uno (o Genuino Uno), con una descrizione completa di tutte le sue parti e una panoramica di tutte le schede esistenti. Il Capitolo 3 è molto operativo e descrive l'installazione e la prima configurazione di Arduino. Nel Capitolo 4 troverete un piccolo corso di programmazione, perché molti aspiranti maker non sono anche dei programmatori ed è importante, per usare Arduino, saper scrivere dei programmi. Il Capitolo 5 presenta le funzioni e i comandi fondamentali di Arduino, che sono poi approfonditi nel Capitolo 6, soffermandosi sempre sugli aspetti pratici del suo utilizzo.

Il Capitolo 7 presenta degli argomenti avanzati: se siete agli inizi, vi consiglio di leggerlo solo se necessario, perché tratta argomenti che non sono usati di frequente e che potrebbero intimorirvi inutilmente. Il Capitolo 8 presenta un'ampia panoramica di sensori con esempi d'impiego, mentre il Capitolo 9 parla di attuatori, transistor e dispositivi che richiedono correnti “importanti” per funzionare. Il Capitolo 10 è dedicato ai meccanismi di comunicazione: dalla porta seriale al GSM. Il Capitolo 11, l'ultimo, è dedicato ad Arduino Yún.

Ho aggiunto un'Appendice “elettronica” per spiegare come leggere le resistenze, perché nel libro è un tema che ho dato per scontato.

È possibile accedere ad approfondimenti, aggiornamenti e contenuti extra visitando www.zepelinmaker.it/manualearduino e arduino.madeformakers.it.

Il manuale di Arduino è nato come libro Open Source e nel giro di pochi mesi ha raggiunto 5000 download: un numero ragguardevole. Vorrei ringraziare tutti coloro che hanno scaricato una copia del libro e mi hanno scritto le loro osservazioni. Un ringraziamento particolare a tutti quelli che hanno contribuito con una donazione per supportare il mio lavoro.

L'autore

Mi chiamo Paolo Aliverti e sono un ingegnere delle telecomunicazioni, artigiano digitale e imprenditore.

Mi sono laureato nel 1999, presso il Politecnico di Milano, con una tesi in robotica e intelligenza artificiale per realizzare un sistema di visione robotica omnidirezionale. Dall'età di dieci anni m'interesso di elettronica e microcomputer. Ho scritto *Il manuale del maker* (tradotto in inglese da Maker Media), *Elettronica per maker* e altri due libri sulla stampa 3D. Scrivo regolarmente su *3D Printing Creative* e inoltre organizzo corsi e workshop su Digital Fabrication, Internet degli Oggetti e Physical Computing. Dal 2008 mi occupo di progettazione e realizzazione di prototipi. Nel 2015 ho lasciato un "posto fisso" presso le ferrovie per dedicarmi alle mie passioni. Appena possibile fuggo su qualche montagna con corda, picca e ramponi.

Avvertenze

La corrente elettrica può essere molto pericolosa: è invisibile e, se non si è consapevoli o sicuri di quello che si fa, si può incorrere in incidenti gravi o mortali. Non utilizzate mai per i vostri esperimenti la tensione di rete a 220 volt. Impiegate solo pile o batterie, prestando comunque la massima attenzione.

Molto tempo fa ero a Roma per lavorare ai robot di RoboCup-99. La squadra del Politecnico di Milano aveva un robot di nome Rullit che giocava nella Middle Size League. Rullit era un robot abbastanza pesante, alimentato da alcuni pacchi di batterie a 12 volt del peso di qualche chilogrammo. Dopo molte ore di programmazione ero sfinito: collegando l'alimentazione del robot invertii i cavi rossi con quelli neri. Ci fu un botto con una piccola esplosione che bucò il tappeto verde del campo da gioco!

Se non siete sicuri o avete dubbi, chiedete a un esperto, a un amico, a un elettricista... Su Internet, anche su Facebook, si trovano molti siti e gruppi (anche se non è facile capire se una persona sia veramente esperta).

Né io né l'editore possiamo assumerci nessuna responsabilità per gli esiti che potrebbero avere gli esperimenti descritti in questo libro. Non possiamo rendere conto per incidenti o danni subiti da cose, persone e animali che potrebbero verificarsi durante gli esperimenti che condurrete.

Il mondo di Arduino

Con una piccola **scheda elettronica** nata in Italia, chiunque può realizzare numerosi circuiti elettronici. **Arduino** o **Genuino** possono essere usati per il funzionamento di **stampanti 3D, droni, prototipi interattivi**. Utilizzarli non richiede particolari competenze e un'intera community è pronta ad aiutarvi.

Quando non avevo ancora acquistato il mio primo Arduino, nel 2008, lessi *Getting Started with Arduino* di Massimo Banzi. Il libro mi piacque molto, perché ritrovai diversi aspetti in comune con l'autore che era anche il padre della famosa scheda elettronica.

Anch'io iniziai da bambino a interessarmi di elettronica e microcomputer. Negli anni '80 possedevo uno Spectrum 48K. Quando aprii la scatola, all'interno vi trovai anche un manuale di programmazione, perché a quei tempi era naturale programmare un computer. Erano gli anni delle riviste di elettronica in edicola, dei kit e dei negozietti di componenti per appassionati. Avevo circa una decina di anni quando incominciai a costruire circuiti elettronici traducendo dei segni sulla carta in oggetti fisici. Non so come imparai. Mi aiutarono due libri che conservo ancora oggi: *L'Elettrotecnica figurata* e *Il manuale dell'inventore*. Erano libri illustrati, adatti alla mia età e che mi furono molto utili. Imparai a distinguere resistenze e condensatori grazie a quei libri e a tanti esperimenti. Avevo un piccolo laboratorio nello scantinato della casa di Ceriano Laghetto: era lo studio di mio padre, che avevo completamente occupato e riempito con tutti i

miei fili e computer. Passavo lì dentro, nel mio piccolo “bunker”, giornate intere... e a volte anche le notti.

Nonostante i miei studi in ingegneria delle telecomunicazioni al Politecnico di Milano, misi da parte il laboratorio per parecchi anni, fino a circa il 2006, quando mi capitò di leggere *FAB* di Neil Gershenfeld, che riaccese la voglia di costruire circuiti elettronici. In una ventina d’anni molto era cambiato: nuove potenti tecnologie erano ormai da tempo degli standard di mercato, tante altre cose si erano semplificate e i prezzi dell’elettronica erano scesi drasticamente.

In questo lungo periodo ho assistito anche all’esplosione di Internet. Ricordo le mie prime “navigazioni” durante le pause tra le lezioni del Politecnico: le sessioni erano in FTP e consistevano in esplorazioni manuali di enormi archivi online. Gli indirizzi si passavano a voce; era come navigare bendati. Qualche anno dopo comparve il browser testuale Lynx per utilizzare il protocollo HTTP: una grandissima novità.

Oggi Internet è diventato un mezzo di comunicazione comune e diffuso ed è di enorme aiuto per chi vuole imparare qualcosa o trovare informazioni.

Nel mio laboratorio di Ceriano Laghetto, le uniche fonti disponibili erano le riviste che conservavo in grandi pile ordinate e che conoscevo a memoria, oltre ad alcuni libri che mi ero fatto regalare e che descrivevano le caratteristiche dei transistor e dei circuiti integrati. Quando possibile, recuperavo componenti smontando elettrodomestici e il più delle volte mi ritrovavo tra le mani parti di cui non conoscevo nulla. Oggi basta digitare la sigla di un qualsiasi componente in Google per avere un datasheet (cioè un foglio d’istruzioni). Su Internet possiamo trovare anche tutti gli schemi elettronici che desideriamo e tutorial e guide per apprendere qualsiasi nozione teorica e pratica. La vita per un “hobbista elettronico” moderno è molto più semplice, anche se è un personaggio quasi estinto. Coloro che s’interessano di questa materia, infatti, oggi sono i *maker*.

Nel 2008 inventai un’App per partecipare a un concorso della Provincia di Milano. L’App serviva agli sportivi per monitorare i loro allenamenti e per proporre loro delle sfide reali o virtuali con altri *runner*. Per evitare alle persone di portarsi il telefono, avevo ideato un aggeggio hardware con GPS, in grado di registrare i dati: un data logger. Per realizzare il prototipo acquistai il mio primo Arduino, con l’intento di collegargli i sensori necessari.

Restai stupito dalla semplicità con cui riuscii a programmarlo la prima volta. Era sufficiente collegare la scheda al PC con un cavo USB, digitare qualche tasto e caricare il tutto premendo un pulsante.

Da allora ho utilizzato Arduino per molti altri prototipi e progetti. Nel 2011 ho iniziato a tenere corsi per imparare a usarlo. Credo fossi tra i primi a offrire a Milano dei corsi come *L’elettronica della sciura Maria* e *L’ABC di Arduino*.

La storia

Arduino¹ nasce presso l'Interaction Design Institute di Ivrea, un istituto di formazione avanzata, postuniversitario, fondato da Olivetti e Telecom. Nasce in un luogo particolare: in una città che ha visto nascere (e tramontare) la grande industria elettronica italiana, che per un momento ci ha fatto sognare in grande. Purtroppo, in Italia è rimasto poco di quelle grandi aziende e di quell'ingegno che sapevano competere con le potenze mondiali e con le multinazionali. Abbiamo inventato il personal computer, siamo stati competitor di colossi come IBM, ma poi qualcosa si è rotto: il nostro sistema politico ed economico ha consumato lentamente quanto costruito da quelle importanti aziende nate nel dopoguerra e gli imprenditori hanno lasciato il posto a manager specializzati in finanza che le hanno spolpate.

Non so bene come sia andata la vicenda, ma si dice che nel 2003 Hernando Barragán abbia sviluppato a Ivrea un progetto originale per avvicinare designer e creativi all'elettronica. Sarebbe meglio dire: per facilitare l'utilizzo dell'elettronica per la realizzazione di prodotti innovativi e interattivi. Il lavoro di Barragán si chiamava Wiring ed è un progetto ancora attivo. Wiring è una piccola scheda elettronica dotata di un chip che può essere programmato semplicemente con un computer e un programma derivato da Processing (un editor per semplificare la programmazione e l'apprendimento del linguaggio Java). Processing è un progetto open source che è stato modificato per programmare Wiring e per funzionare con il linguaggio C richiesto dalla scheda elettronica. Wiring e Processing "modificato" costituiscono un sistema, o framework, per la progettazione rapida di circuiti elettronici. La scrittura del codice non è fatta in C puro, ma con una forma semplificata, nascondendo i dettagli della programmazione a basso livello: tutto per facilitarne l'utilizzo da parte di persone inesperte.

Massimo Banzi si interessò al lavoro di Barragán e nel 2005 decise di sviluppare una nuova scheda partendo dall'idea molto originale di Wiring, cercando di semplificare maggiormente l'utilizzo del sistema e abbassando il costo della scheda. Il nuovo progetto fu battezzato Arduino, il nome del bar di Ivrea, dove il team del progetto si trovava per bere l'aperitivo. Il team che realizzò Arduino era composto da Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. Da allora il progetto ha riportato notevole successo in tutto il mondo dei maker e non solo. La piccola schedina di Ivrea è stata adottata per innumerevoli progetti e ha permesso la realizzazione di cose fino a poco tempo prima impensabili, soprattutto da parte di persone che non avevano conoscenze elettroniche.

¹ La storia riportata è frutto della mia immaginazione e dalle fonti reperibili in Rete. Non ho mai potuto intervistare i diretti interessati e spero che quanto citato sia il più attendibile possibile. Mi perdonino Massimo Banzi e Hernando Barragán se ho scritto qualche bestialità!

Nasce così Arduino, una piccola scheda con un microcontrollore, che chiunque può imparare a utilizzare in breve tempo per realizzare circuiti elettronici interattivi. Per programmare Arduino è sufficiente collegarlo al proprio computer con un cavo USB, scrivere qualche istruzione, collegare qualche componente elettronico e premere il tasto Upload. Si procede per tentativi ed errori, correggendo il programma di volta in volta. Le istruzioni sono abbastanza semplici e comprensibili. Il software di programmazione, chiamato anche IDE (Ambiente Integrato di Sviluppo), contiene numerosi esempi da cui prendere spunto per scrivere i propri listati. Per facilitare la costruzione dei circuiti elettrici, sono state inventate delle piccole schede chiamate *shield* che s'impilano su Arduino. Esistono shield con display, con un modulo Bluetooth, pulsanti, controllori per motori, GPS, schede di rete, moduli audio ecc. Senza neppure spelare un filo elettrico, potete costruire un "panino" di shield infilate l'una sopra l'altra e realizzare dispositivi anche molto sofisticati.

Esistono kit modulari formati da una shield madre da collegare ad Arduino e tante piccole schedine e cavetti che si connettono per comporre i circuiti che servono. Ci sono schedine con pulsanti, sensori di vario tipo, motori, display, LED, relè.

Arduino ha successo perché è semplice e funziona sempre. Perché trovate spesso quello che vi serve, senza dovervi scervellare troppo. Perché è un prodotto ben fatto e continuamente curato. La scheda è stata sempre prodotta in Italia e continua a esserlo e si può considerare un vero prodotto *Made in Italy*.

Il progetto didattico si è trasformato in prodotto mantenendo la caratteristica open source. Non c'è nulla di segreto o di nascosto. Gli schemi dell'hardware sono liberamente scaricabili da Internet, così come tutto il software necessario per costruire in casa, autonomamente, il proprio Arduino.

Le aziende sono spesso abbastanza paranoiche sul tema sicurezza e se possibile terrebbero tutto sotto chiave. È inimmaginabile diffondere su Internet i piani del proprio prodotto di punta. Arduino ha dimostrato che con le opportune accortezze si possono creare aziende di successo basate su prodotti open source. La tutela è sul marchio e la protezione non sta in un brevetto, ma sulla rispettabilità del nome e del prodotto: rispettabilità costruita online grazie alla collaborazione e al coinvolgimento di migliaia di maker.

In seguito a una recente controversia sul marchio Arduino, in tutto il mondo esclusi gli Stati Uniti la scheda è venduta con il nome di "Genuino", anche se fondamentalmente è identica. Negli USA è invece commercializzata con il nome "Arduino". Anche in Italia potete acquistare Arduino, ma è la scheda prodotta dalla storica Smart Projects, ora Arduino S.r.l. Questo induce indubbiamente un po' di confusione, ma tutto ciò che potete trovare nel presente libro su Arduino è applicabile nello stesso modo alle nuove schede Genuino.

Microcontrollori e microprocessori

Che cos'è un microcontrollore? Prima dovrete sapere che cos'è un microprocessore: un circuito integrato (anche chiamato *chip*), specializzato nell'esecuzione di operazioni matematiche, nel trasferimento e nella manipolazione d'informazioni. Questi chip sono il cervello dei nostri computer e per funzionare hanno bisogno di memorie, dischi, periferiche di vario tipo, mouse, tastiera, display. Il primo microprocessore è stato costruito attorno agli anni '70 da Intel. Il circuito prodotto da Intel si chiamava 4004, conteneva qualche manciata di transistor e sapeva svolgere semplici operazioni matematiche a quattro bit, con frequenze di qualche kHz. I microprocessori di oggi sono delle grandi metropoli se paragonate al 4004: hanno centinaia di migliaia di transistor e lavorano a frequenze di GHz.

Un microcontrollore è un chip che contiene un piccolo microprocessore e il minimo indispensabile per il suo funzionamento: della memoria, qualcosa che possa funzionare da disco (*flash memory*), una memoria a lungo termine (EEPROM), un generatore di frequenza, dei convertitori analogico-digitali e delle semplici periferiche per permettergli di interagire con il mondo. Molti microcontrollori hanno potenza limitata, ma anche costi molto bassi. Per questo motivo li troviamo in molti dispositivi che utilizziamo quotidianamente. Grazie a una serie d'innovazioni introdotte da ATMEL e Microchip attorno agli anni '90, il loro utilizzo è diventato molto semplice, contribuendo alla loro diffusione. Un microcontrollore moderno non richiede speciali apparati per essere programmato e il firmware può essere trasferito al suo interno anche se è già stato saldato su un circuito stampato. Prima di queste innovazioni, per utilizzare un microcontrollore era necessario esporlo a dei raggi UV per cancellarlo e poi programmarlo con kit di sviluppo molto costosi. I microcontrollori moderni hanno la possibilità di ospitare un piccolo programma, chiamato *bootloader*, residente in un'area di memoria speciale. Il bootloader è un programma che può scrivere direttamente le istruzioni nella memoria del chip. In questo modo non serve un programmatore particolarmente complicato: con una semplice porta seriale è possibile programmare il chip con il software che abbiamo scritto.

Spesso sentirete parlare di System On Chip. Con questo termine s'indicano dei chip ancora più complicati e completi di un microcontrollore. Sono circuiti integrati che contengono, oltre a un microprocessore, anche la circuiteria per schede di rete o video; sono molto simili a un computer vero e proprio.

Esistono diverse famiglie di microcontrollori, ognuna contraddistinta da un proprio nome e dall'organizzazione tipica dei suoi circuiti interni, cioè l'architettura del chip. La programmazione di questi chip era fatta in codice macchina, scrivendo lunghi programmi che poi erano convertiti in sequenze di numeri. Inizialmente, il numero di comandi e di operazioni era limitato, si potevano eseguire semplici operazioni logiche,

trasferimenti di dati e test. Con l'aumentare della complessità dei chip, i progettisti iniziarono a inserire delle operazioni più specifiche e sempre più complesse.

Questi chip con un set d'istruzioni molto numeroso si chiamano CISC (Complex Instruction Set Computer). Sono circuiti di grosse dimensioni, molto costosi da progettare e da realizzare. I programmatori però usavano raramente tutte queste istruzioni complesse e a volte compiere la stessa operazione in più passaggi e con comandi base era addirittura più veloce. Per questo motivo sono nati i chip RISC (Reduced Instruction Set Computer), in cui il set d'istruzioni è stato ridotto al minimo indispensabile.

I più noti processori RISC sono i PIC, gli AVR, gli ARM o gli SPARC. Il chip utilizzato da Arduino Uno è un ATmega328 della famiglia AVR. Avrete spesso sentito nominare i chip ARM (Advanced RISC Machine), che sono circuiti impiegati nella realizzazione di telefoni, tablet e dispositivi portatili con bassi consumi e ottime prestazioni.

Shield

Arduino è modulare. Sulla scheda elettronica sono presenti dei piccoli connettori in cui collegare fili elettrici o componenti. Questi connettori presentano delle serie di sei oppure otto fori, chiamati anche *pin*. È possibile infilare su questi connettori delle schede di espansione che rispettino la stessa disposizione di contatti.

Le shield sono schede specializzate per realizzare specifiche funzioni. Per utilizzarne una non dovete fare altro che infilarla sopra ad Arduino, creando così un "panino" di schede. Si possono impilare più schede riuscendole a controllare tutte senza conflitti perché ogni scheda occupa un certo numero di pin. Le shield sono sviluppate e prodotte da Arduino, ma anche altri famosi produttori di elettronica per maker hanno creato delle schede compatibili: l'importante è rispettare la piedinatura dei pin. Abbiamo shield con display LCD e pulsanti, con Ethernet, per il WiFi, per il Bluetooth, per controllare motori elettrici, con il GPS, con il GSM, con display e-ink, data logger con SD card e clock e perfino shield per collegare dispositivi medicali. Non dovete fare altro che acquistare quello che vi serve e infilarlo sul vostro Arduino.

Un'altra possibilità per semplificare lo sviluppo elettronico di circuiti è di utilizzare kit modulari come TinkerKit o Grove. Questi sistemi sono costituiti da una shield dotata di molti piccoli connettori. La shield s'infilare su Arduino e ci sono molte piccole schedine che si possono collegare alla scheda madre. I collegamenti si fanno con dei semplici cavetti dotati di connettori standard. Ogni piccola schedina monta un singolo componente o svolge una semplice funzione. Avete a disposizione schede con un semplice LED, un pulsante, schede con un relè, un potenziometro, un sensore di temperatura, per il gas o per la luce ecc. Per comporre i circuiti dovete semplicemente cablare alla scheda madre quello che vi serve.

Maker & Arduino

I maker hanno subito apprezzato la versatilità di Arduino e l'hanno utilizzato per realizzare una moltitudine di progetti, condividendo i risultati raggiunti, gli schemi elettrici e il software. Alcuni di questi progetti sono molto noti su Internet, altri si sono trasformati in startup. Vi presento qui alcuni lavori che mi hanno particolarmente colpito. In ciascuno, anche qualora notiate una certa aria "da garage" e da prototipo, apprezzate lo spirito di condivisione e percepite la passione di chi vi ha lavorato. Questi maker (tra cui il sottoscritto) impiegano ore e ore del loro tempo per lavorare a progetti che non sempre hanno fini commerciali. Le luci delle nostre cantine (o della mia mansarda) stanno accese fino a tardi per consentirci di lavorare a quello che ci sta veramente a cuore.

Stampanti 3D

Arduino ha avuto un ruolo importante nello sviluppo delle stampanti 3D open source: la maggior parte delle schede elettroniche utilizzate oggi nelle stampanti deriva da Arduino o da una delle sue numerose varianti. Per realizzare una stampante con Arduino sono necessari alcuni particolari circuiti di pilotaggio per dei motori passo-passo chiamati *driver*, una manciata di transistor e termistori per controllare la temperatura. Molte delle schede oggi in commercio, come le RAMPS, derivano da una scheda Arduino modificata, con integrata l'elettronica necessaria per il funzionamento della stampante.

CNC e laser cutter

Se con Arduino posso costruire una stampante 3D, perché non una laser cutter o una macchina CNC? In effetti, c'è poca differenza e l'elettronica potrebbe, in alcuni casi, essere anche più semplice. Una CNC non ha delle parti che si riscaldano e al massimo prevede un controllo di velocità per un utensile. Una semplice fresa CNC potrebbe quindi essere costruita con tre motori per pilotare gli assi X, Y e Z e un controllo di velocità per l'utensile. Il progetto Lasersaur utilizza una scheda da collegare ad Arduino per pilotare una macchina open source per il taglio laser.

Droni

I droni sono piccoli elicotteri multirotores (cioè dotati di più eliche) capaci di volare in modo molto stabile e controllato. Serve comunque una certa "intelligenza" sia per il pilotaggio dei motori sia per il controllo della stabilità dell'elicottero e Arduino è perfetto per gestire i motori, l'assetto, leggere un accelerometro e altri sensori di orientamento, oltre che per ricevere comandi da una radiotrasmittente.

Anche in questo caso, dopo i primi prototipi creati con delle semplici schede Arduino, sono comparse delle *board* specializzate ma sempre compatibili con la piattaforma originale. In questo modo, è possibile intervenire sul programma di pilotaggio, modificando a piacere parametri e comportamenti.

ArduFonino

Con Arduino è possibile realizzare un telefonino personalizzato. Non potrà essere un telefono di dimensioni ridotte, ma è totalmente personalizzabile. Uno dei progetti che si possono trovare in Rete si chiama ArduFonino ed è stato realizzato da Davide Aloisi (<http://www.davidealoisi.it/>). Sul suo sito racconta così il progetto: "ArduFonino è un cellulare open source creato con Arduino, capace di effettuare e ricevere chiamate e di inviare e ricevere messaggi, inoltre con la realizzazione di questo progetto ho voluto gettare le basi per la costruzione di un sistema telefonico elementare. L'idea mi è venuta per semplice curiosità, perché volevo fare qualcosa che nessuno avesse mai fatto, dimostrando a tutti che volere è potere".

Il telefonino utilizza Arduino, una shield GSM con alloggiamento per la SIM e una shield custom con display, tastiera, microfono e altoparlante. È molto semplice collegare Arduino ad altri dispositivi o sistemi utilizzando delle shield Bluetooth, WiFi, GSM.

Robotica

Per utilizzare al meglio tutte le conoscenze su Arduino non c'è nulla di più emozionante che applicarle nella costruzione di un robot. Per un progetto del genere, Arduino può svolgere la funzione di controllo, leggere informazioni dai sensori e trasmettere i movimenti a degli attuatori, come motori e servocomandi. Oggi è semplice costruirsi un robot DIY (Do It Yourself) ovvero "Fai da te", soprattutto se avete a disposizione una stampante 3D per realizzare tutte le parti meccaniche. In alternativa, esistono parecchi kit "meccanici" che si possono acquistare e completare aggiungendo sensori, motori e altri dispositivi. Il robot può essere controllato da remoto via radio, ma la vera sfida è di renderlo autonomo, programmando l'intelligenza artificiale necessaria per fargli prendere le corrette decisioni e per farlo muovere nel mondo in autonomia, evitando ostacoli e risolvendo problemi. Arduino ha creato uno speciale kit robotico: un robot completo formato da sue schede, di cui una con due ruote motorizzate. Non avete che da aggiungere qualche sensore e iniziare a programmare.

Garduino

Sono veramente tanti i progetti in cui è stato utilizzato Arduino. Basta fare qualche ricerca su Google per restare stupiti dalla fantasia e dall'ingegnosità di molte realizza-

zioni. Qualcuno ha pensato di utilizzarlo per badare alla cura delle piante domestiche: la piccola scheda può misurare l'umidità del terreno, la temperatura, l'esposizione solare e può irrigare le piante avvisandoci con una mail. Esistono tanti progetti per controllare il proprio giardinetto: il più famoso è Garduino, oggi noto come Growerbot, un sistema per irrigare le piante quando necessario, che può anche accendere delle luci supplementari se il sole non è sufficiente e che ci può avvisare quando le condizioni ambientali non sono adeguate, perché fa troppo caldo o troppo freddo.

L'unico limite è la fantasia, perché poi, in effetti, realizzare il circuito non è così complicato. Arduino può essere utilizzato anche per riparare oggetti che non funzionano più. Mi è capitato di usarlo per aggiustare il boiler di un camper: la ditta produttrice è fallita dieci anni fa e non si trovano più le parti di ricambio, quindi ho sostituito la scheda di controllo della caldaia con Arduino, collegandolo al sensore di fiamma, ai sensori di temperatura dell'acqua e all'elettrovalvola per il controllo del gas.

Siti di riferimento

Il principale sito di riferimento è <http://www.arduino.cc>, anche se recentemente, in seguito ad alcuni problemi legali dei fondatori, è stato aperto un sito "clone" <http://www.arduino.org>. La disputa è aperta e non si sa come andrà a finire, ma il sito originale resta arduino.cc. Da qui potete accedere al forum e a numerose risorse online, come tutorial, istruzioni per la programmazione, esempi di circuiti e informazioni di utilizzo delle schede.

Una ricerca su Google ci farà scoprire numerose risorse, siti, blog, tutorial, video, dispense e libri. C'è solo l'imbarazzo della scelta. Se potete, però, chiudete il vostro computer e visitate un makerspace o uno degli Arduino User Group sparsi in tutt'Italia: probabilmente, ne esiste uno anche nella vostra città. Troverete tanti appassionati come voi disposti a fare una chiacchierata e a condividere le loro esperienze. Gli AUG organizzano periodicamente eventi e incontri per illustrare nuovi argomenti, progetti e approfondimenti. Di solito, queste comunità hanno anche una forte presenza online sui social media come Facebook, Twitter e Google+.