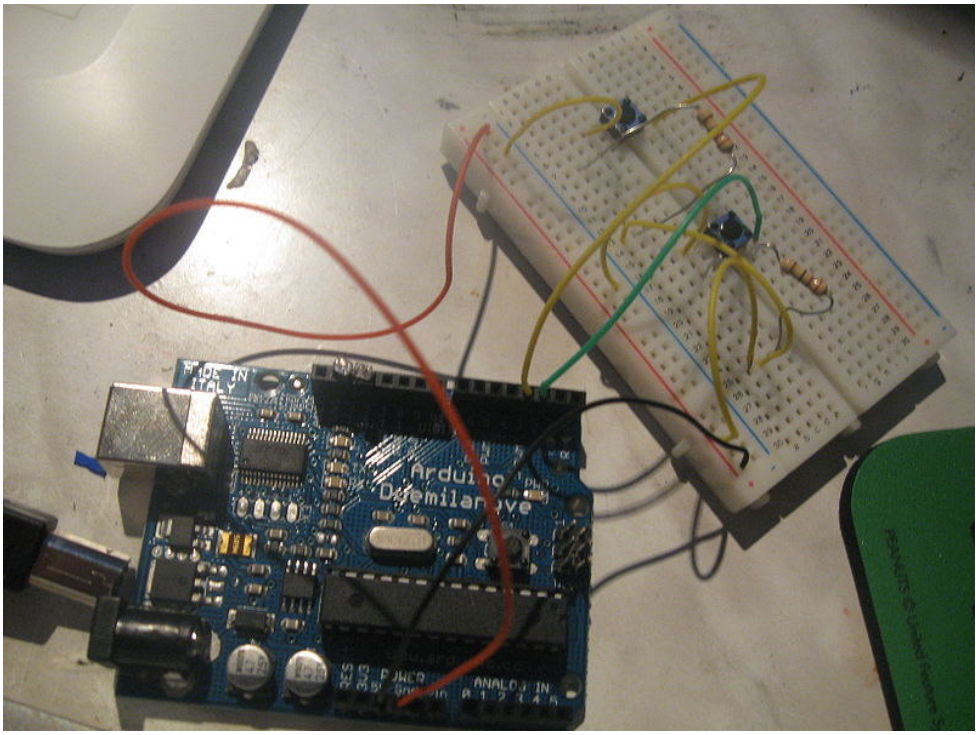
Che cosa è Arduino?
(tratto da Wikipedia)

Arduino è un sistema open source che permette la realizzazione rapida e l'apprendimento veloce dei principi fondamentali dell'elettronica e della programmazione. È composto da una piattaforma hardware sviluppata presso l'Interaction Design Institute, un istituto con sede ad Ivrea, fondato da Olivetti e Telecom Italia[2]. Il nome della scheda deriva da quello di un bar di Ivrea frequentato da alcuni dei fondatori del progetto[3].



Questa si basa su un circuito stampato che integra un microcontrollore con PIN connessi alle porte di I/O, un regolatore di tensione e, quando necessario, un'interfaccia USB che permette la comunicazione con il computer. A questo hardware viene affiancato un ambiente di sviluppo integrato (IDE) multipiattaforma (Linux, Apple Macintosh e Windows). Questo software permette anche ai novizi di scrivere programmi con un linguaggio semplice e intuitivo derivato da C e C++ chiamato Wiring, liberamente scaricabile e modificabile.

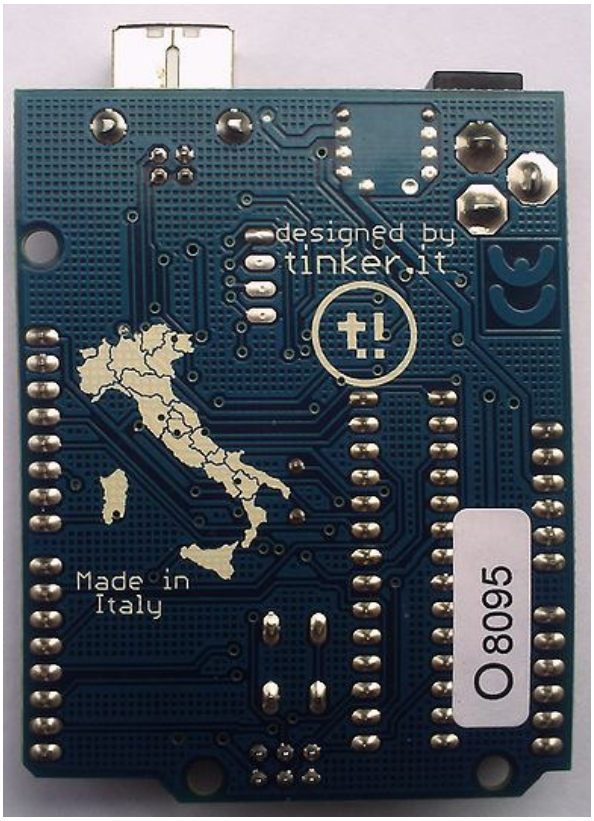
Arduino può essere utilizzato per lo sviluppo di oggetti interattivi *stand-alone* ma può anche interagire, tramite collegamento, con software residenti su computer, come Adobe Flash, Processing, Max/MSP, Pure Data, SuperCollider, Vvvv.

La piattaforma hardware Arduino è spesso distribuita agli hobbisti in versione pre-assemblata, acquistabile in internet o in negozi specializzati. La particolarità del progetto è che le informazioni sull'hardware e soprattutto i progetti sono disponibili per chiunque: si tratta quindi di un hardware open source, distribuito nei termini della licenza Creative Commons Attribution-ShareAlike 2.5[4]. In questo modo, chi lo desideri può legalmente auto-costruirsi un clone di Arduino o derivarne una versione modificata, scaricando gratuitamente lo schema elettrico e l'elenco dei componenti elettronici necessari[4]. Questa possibilità ha consentito lo sviluppo di prodotti Arduino compatibili da parte di piccole e medie aziende in tutto il mondo, e infatti oggi è possibile scegliere tra molte schede Arduino compatibili. Ciò che accomuna questi prodotti relativi all'elettronica sperimentale e sviluppo è il

codice sorgente per l'Ambiente di sviluppo integrato e la libreria residente che sono resi disponibili, e concessi in uso, secondo i termini legali di una licenza libera, GPLv2.

Grazie alla base software comune, ideata dai creatori del progetto, per la comunità Arduino è stato possibile sviluppare programmi per connettere a questo hardware più o meno qualsiasi oggetto elettronico, computer, sensore, display o attuatore. Dopo anni di sperimentazione è oggi possibile fruire di un database di informazioni vastissimo.

Il team di Arduino è composto da Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, e David Mellis. Il progetto prese avvio in Italia a Ivrea nel 2005, con lo scopo di rendere disponibile, a progetti di Interaction design realizzati da studenti, un dispositivo per il controllo che fosse più economico rispetto ai sistemi di *prototipazione* allora disponibili. I progettisti riuscirono a creare una piattaforma di semplice utilizzo ma che, al tempo stesso, permetteva una significativa riduzione dei costi rispetto ad altri prodotti disponibili sul mercato. A ottobre 2008 in tutto il mondo erano già stati venduti più di 50.000 esemplari di Arduino.



L'hardware originale Arduino è realizzato dalla italiana *Smart Projects*. Alcune schede Arduino sono state progettate dalla statunitense *SparkFun Electronics*.

Una scheda Arduino tipica consiste in un microcontroller a 8-bit AVR prodotto dalla Atmel, con l'aggiunta di componenti complementari per facilitarne l'incorporazione in altri circuiti. In queste schede sono usati chip della serie megaAVR - nello specifico i modelli ATmega8, ATmega168, ATmega328, ATmega1280 e ATmega2560.

Molte schede includono un regolatore lineare di tensione a 5-volt e un oscillatore a cristallo a 16 MHz, sebbene alcune implementazioni, come ad esempio la piccola *LilyPad*[5], abbiano un clock di 8 MHz e facciano a meno dello stabilizzatore di tensione.

Fino a oggi sono state commercializzate dodici versioni dell'hardware Arduino.[4]

1. Serial Arduino, programmata con una porta seriale DB9. Fa uso del microcontroller ATmega8
2. Arduino Extreme, con interfaccia di programmazione USB, facente uso del chip ATmega8
3. Arduino Mini, una versione in miniatura facente uso di un ATmega168 a montaggio superficiale
4. Arduino Nano, una versione ancora più piccola della Mini, utilizzante lo stesso controller ATmega168 SMD e alimentata tramite USB
5. LilyPad Arduino, un progetto minimalista (scheda circolare dal diametro di 50mm, per circa 8mm di spessore), per applicazione su indumenti, con lo stesso ATmega168 in versione SMD[5]
6. Arduino NG, con un'interfaccia USB per programmare e usare un ATmega8
7. Arduino NG plus, con interfaccia di programmazione USB, con un ATmega168
8. Arduino BT, con interfaccia di programmazione Bluetooth e con un ATmega168
9. Arduino Diecimila, con interfaccia di programmazione USB e con un ATmega168 in un package DIL28
10. Arduino Duemilanove, facente uso del chip Atmega168 (o Atmega328 nelle versioni più recenti) e alimentata in corrente continua tramite USB, con commutazione automatica tra le sorgenti di alimentazione
11. Arduino Mega, che fa uso di un ATmega1280 a montaggio superficiale per I/O e memoria addizionale.
12. Arduino Uno, evoluzione della Duemilanove, con un differente chip, programmabile e più economico, dedicato alla conversione USB-seriale.
13. Arduino Mega2560, che fa uso di un ATmega2560 ed è un'evoluzione dell'Arduino Mega.

La programmazione del microcontroller

Inoltre, il microcontroller della scheda in alcuni case è pre-programmato con un bootloader che semplifica il caricamento dei programmi sulla memoria flash incorporata nel chip.

A livello concettuale, tutte le schede sono programmate attraverso un porta seriale RS-232, ma il modo in cui questa funzionalità è implementata nell'hardware varia da versione a versione. Le schede seriali Arduino contengono un semplice circuito inverter che permette la conversione tra il livello della RS-232 e il livello dei segnali TTL.

Le versioni attuali di Arduino sono gestite via USB: l'ultima versione, la **Uno**, utilizza un microcontrollore Atmega8U2 programmato come convertitore USB-seriale mentre le precedenti versioni *Diecimila* e *Duemilanove* usavano chip adattatori USB-seriale, come gli FT232 di FTDI. Alcune varianti, come la Arduino Mini e la versione non ufficiale *Boarduino*, usano una scheda o un cavo adattatore USB-seriale staccabile.

Funzionalità di Input/Output

Per implementare il comportamento interattivo, Arduino è fornita di funzionalità di Input/Output, grazie alle quali essa riceve i segnali raccolti da sensori esterni. In base a tali valori, il comportamento della scheda è gestito dal microcontroller, in base alle decisioni determinate dal particolare programma in esecuzione in quel momento sulla scheda. L'interazione con l'esterno avviene attraverso attuatori pilotati dal programma attraverso i canali di output in dotazione.

Connettori I/O

A tale scopo, Arduino è dotata di molti dei connettori di Input/Output per microcontroller in uso su altri circuiti. Tutti i pin di I/O sono collocati sulla parte superiore della scheda, mediante connettori femmina da 0,1". Inoltre, sono disponibili commercialmente molte schede applicative plug-in, note come "shields".

Le schede *Barebones* e *Boarduino*, due cloni compatibili con la Arduino, sono dotate di connettori maschio sul lato inferiore del circuito in modo da poter essere connesse a una breadboard senza necessità di effettuare saldature.

I/O digitale

La scheda Arduino offre 14 connettori per l'I/O digitale (numerati da 0 a 13). La direzione di funzionamento, input o output, è decisa dallo *sketch* programmato sull'IDE.

Sei dei canali I/O possono produrre segnali Pulse-width modulation (PWM). Attraverso i segnali PWM è possibile, ad esempio, regolare l'intensità di luminosità di un LED o la velocità di rotazione di un motorino elettrico[6]. L'hardware di tre dei pin di I/O (9, 10 e 11) implementa la possibilità di gestirli direttamente attraverso la funzione `analogWrite()`, che permette di controllare la PWM del segnale in uscita in maniera efficiente, senza dover eseguire linee di codice appositamente predisposte[7]. La funzione accetta due parametri, il primo dei quali è il pin pilotato mentre il secondo rappresenta l'intensità della modulazione (espressa su una scala da 0 a 255): così, ad esempio, `analogWrite(9, 128)` attiverà un led collegato al pin 9 al 50% della sua luminosità[8].

I/O analogico

Sono presenti altri 6 connettori specificamente dedicati a ingressi di segnali analogici, cioè valori di tensione letti da sensori esterni i cui valori, fino a un massimo di 5 Volt, sono convertiti in 1024 livelli discreti (da 0 a 1023). Questi 6 connettori possono essere riprogrammati (sempre dal codice dello *sketch* sull'IDE) per funzionare come normali entrate/uscite digitali.

Alimentazione elettrica

L'alimentazione della scheda può avvenire attraverso la porta USB del computer, o attraverso la maggior parte degli alimentatori USB, oppure attraverso un adattatore in corrente continua a 9 volt, con connettore cilindrico (diametro 2,1 mm e positivo centrale). In quest'ultimo caso, la scheda commuta automaticamente sull'alimentazione esterna quando il connettore dell'alimentatore esterno è inserito, mentre commuta autonomamente sull'alimentazione USB in caso di disconnessione del connettore. La Arduino-NG e la Arduino Diecimila, versioni meno recenti, necessitano di essere commutate a mano, azionando uno switch ubicato tra la porta USB e l'ingresso dell'alimentazione esterna.

Arduino	Processore	Flash KiB	EEPROM KiB	SRAM KiB	Pin di I/O Digitale	di cui con PWM	Pin di Input Analogico	Tipo Interfaccia USB	Dimensioni in pollici	Dimensioni in mm
Diecimila	ATmega168	16	0.5	1	14	6	6	FTDI	2.7 in × 2.1 in	68.6 mm × 53.3 mm
Due[9]	ATMEL SAM3U	256		50			16			
Duemilanove	ATmega168/328P	16/32	0.5/1	1/2	14	6	6	FTDI	2.7 in × 2.1 in	68.6 mm × 53.3 mm
Uno	ATmega328P	32	1	2	14	6	6	ATmega8U2	2.7 in × 2.1 in	68.6 mm × 53.3 mm
Leonardo	Atmega32u4	32	1	2	14	6	12	Atmega32u4 integrato	2.7 in × 2.1 in	68.6 mm × 53.3 mm
Mega	ATmega1280	128	4	8	54	14	16	FTDI	4 in × 2.1 in	101.6 mm × 53.3 mm
Mega2560	ATmega2560	256	4	8	54	14	16	ATmega8U2	4 in × 2.1 in	101.6 mm × 53.3 mm
Fio	ATmega328P	32	1	2	14	6	8	Nessuno	1.6 in × 1.1 in	40.6 mm × 27.9 mm
Nano	ATmega168 o ATmega328	16/32	0.5/1	1/2	14	6	8	FTDI	1.70 in × 0.73 in	43 mm × 18 mm
LilyPad	ATmega168V o ATmega328V	16	0.5	1	14	6	6	Nessuno	2 in ⌀	50 mm ⌀

Schede Arduino compatibili

La quantità enorme e l'estrema variabilità d'uso e di componenti, rende difficile definire univocamente una scheda Arduino compatibile. Solitamente contengono un microcontroller a 8 16 o 32 bit AVR, PIC o ARM, con clock variabile tra gli 1 e 96 MHz. Molte incorporano componenti aggiuntivi pensati per i più svariati utilizzi.

La programmazione del microcontroller

La programmazione può avvenire tramite il protocollo ISP e un'altra scheda Arduino utilizzata come programmatore, tramite la usb (in alcuni casi il microcontrollore contiene usb hardware) oppure grazie a un programmatore esterno dedicato.

L'ambiente di sviluppo integrato (IDE) di Arduino è un'applicazione multipiattaforma scritta in Java, ed è derivata dall'IDE creato per il linguaggio di programmazione Processing e per il progetto Wiring. È concepita per iniziare alla programmazione artisti e altri neofiti, che siano a digiuno di pratica nello sviluppo di software. Per permettere la stesura del codice sorgente, l'IDE include un editore di testo dotato inoltre di alcune particolarità, come il syntax highlighting, il controllo delle parentesi, e l'indentazione automatica. L'editor è inoltre in grado di compilare e lanciare il programma eseguibile in una sola passata e con un solo click. In genere non vi è bisogno di creare dei Makefile o far girare programmi dalla riga di comando.

L'ambiente di sviluppo integrato di Arduino è fornito di una libreria software C/C++ chiamata "Wiring" (dall'omonimo progetto Wiring): la disponibilità della libreria rende molto più semplice implementare via software le comuni operazioni di input/output. I programmi di Arduino sono scritti in C/C++, ma all'utilizzatore, per poter creare un file eseguibile, non è richiesto di scrivere un programma in C, ma solo di definire due funzioni:

- `setup()` funzione invocata una sola volta all'inizio di un programma che può essere utilizzata per i settaggi iniziali;
- `loop()` funzione chiamata ripetutamente, la cui esecuzione si interrompe solo con lo spegnimento della scheda.

Esempi di codice

Un tipico esempio di programma per iniziare a far pratica con un microcontroller è quello, molto semplice, che permette l'accensione ripetuta di un LED. Nell'ambiente di sviluppo di Arduino, l'utilizzatore potrebbe farlo con un programmino come il seguente[10] :

```
#define LED_PIN 13
void setup () {
    pinMode (LED_PIN, OUTPUT);      // abilita il pin 13 per l'output digitale
}

void loop () {
    digitalWrite (LED_PIN, HIGH);   // accende il LED
    delay (1000);                   // aspetta 1 secondo (1000 millisecondi)
    digitalWrite (LED_PIN, LOW);    // spegne il LED
    delay (1000);                   // aspetta un secondo
}
```

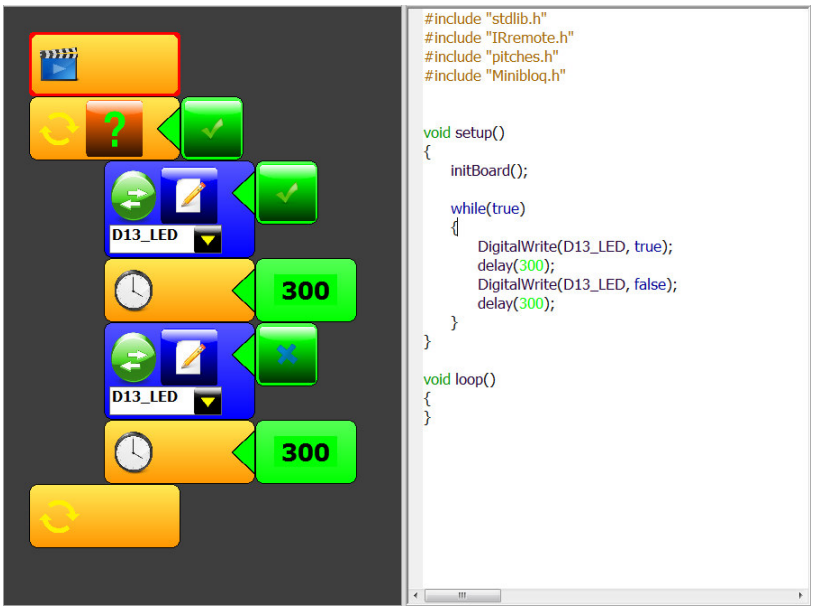
Scritto in questo modo, il codice non verrebbe riconosciuto come programma valido da un compilatore C++; per questo motivo, quando l'utilizzatore seleziona il bottone di comando "Upload to I/O board" (presente nell'IDE), una copia del codice viene automaticamente trascritta in un file temporaneo con l'aggiunta di un extra include header all'inizio del codice e l'implementazione di una semplicissima funzione `main()` in basso:

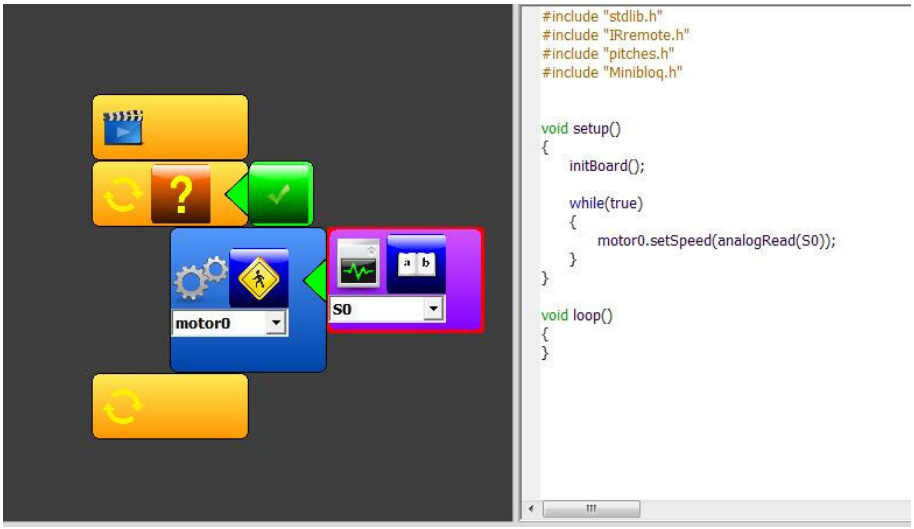
```
#include "WProgram.h"
#define LED_PIN 13

void setup () {
    pinMode (LED_PIN, OUTPUT);      // abilita il pin 13 per l'output digitale
}
void loop () {
    digitalWrite (LED_PIN, HIGH);   // accende il LED
    delay (1000);                   // aspetta 1 secondo (1000 millisecondi)
    digitalWrite (LED_PIN, LOW);    // spegne il LED
    delay (1000);                   // aspetta 1 secondo
}
int main(void)
{
    init();
    setup();
    for (;;)
        loop();
    return 0;
}
```

"WProgram.h" è lo header file principale per la libreria Wiring. La funzione `main()` effettua solo tre chiamate, `init()`,`setup()` e `loop()`; le ultime due sono definite dall'utente, mentre la prima è scritta nella libreria stessa. La funzione `loop()`, come richiesto, è automaticamente nidificata in un loop infinito.

Lo IDE di Arduino usa la GNU toolchain e la AVR Libc per compilare i programmi, mentre usa avrdude per caricarli sulla scheda.

Minibloq



Un esempio alternativo di ambiente integrato dedicato ad Arduino (ma anche ad altre utilizzazioni) è Minibloq, distribuito con licenza libera, che mette a disposizione un particolare ambiente di sviluppo visuale per la programmazione di Arduino, robot, e altri simili dispositivi di physical computing.

Caratteristiche di Minibloq sono l'interfaccia intuitiva, e la capacità di generare il codice a partire da uno schema grafico a blocchi disegnato dall'utente. Per questa sua impostazione intuitiva, il software è indirizzato ad avvicinare principianti e neofiti alla programmazione di microcontroller, in contesti scolastici ed educativi che vanno dalla scuola primaria all'ambiente liceale. L'applicazione, compilata con GCC, si presta a essere utilizzata anche su computer a più basse prestazioni, come netbooks, OLPC XO-1, Intel-Classmate.