



Data: 26-4-2016

pag. 6

rel. 1.0

- **Schede Arduino**

Arduino è una serie di schede di piccole dimensioni con un microcontrollore e circuiteria di contorno, utili per creare rapidamente prototipi di automazione ed anche utile e per scopi hobbistici e didattici. Il nome della scheda deriva da quello di un bar di Ivrea (che richiama a sua volta il nome di Arduino d'Ivrea, Re d'Italia nel 1002) frequentato da alcuni dei fondatori del progetto

Con Arduino si possono realizzare in maniera relativamente rapida e semplice piccoli dispositivi come controllori di luci, di velocità per motori, sensori di luce, temperatura e umidità e molti altri progetti che utilizzano sensori, attuatori e comunicazione con altri dispositivi.

E' a buon mercato, si collega direttamente alla porta USB del computer, e rispetto alle altre schede di sviluppo è molto, molto semplice da configurare e utilizzare.

Ecco le caratteristiche chiave della Scheda **ARDUINO UNO R3**:

1) è un progetto **open source**.

2) , ha un'**interfaccia USB semplice**. Il chip sulla scheda si collega direttamente alla porta USB e viene visto dal computer come una porta seriale virtuale

3) **gestione automatica della tensione**: sarà la scheda a organizzarsi internamente, a noi basta collegare una fonte di alimentazione esterna di 12v al massimo e sarà tutto perfetto; può essere alimentato direttamente dalla porta USB, senza alimentazione esterna aggiuntiva.

4) il **chip ATmega328** costa al dettaglio circa 4-5 euro. Ha innumerevoli funzionalità molto comode come il timer, la gestione della PWM su determinati pin, molteplici modalità di sospensione e altro ancora

5) un **clock al quarzo a 16MHz**, abbastanza veloce per la maggior parte delle applicazioni;

32 KB di memoria flash per la memorizzazione di codice ;

6) **i pin**: 13 digitali e 6 analogici. Con questi piedini colleghiamo il mondo esterno ad Arduino. Sono fondamentali per estenderne le possibilità, basta collegare i dispositivi, i sensori, quello che riusciamo a inventare nelle prese corrispondenti a ciascuno di questi pin e siamo – quasi – a posto.

7)- un **LED montato a bordo** per vedere se la scheda è viva o per fare rapidi test semplici (è collegato direttamente al pin digitale 13).

8) un **pulsante di reset**, per resettare la scheda

- **L'architettura**

Arduino comprende una piattaforma hardware per il **physical computing** sviluppata presso l'Interaction Design Institute (IDI), un istituto di formazione post-dottorale con sede a Ivrea, fondato da Olivetti e Telecom Italia.

Questa si basa su un circuito stampato che integra:

- un **microcontrollore con pin connessi** alle porte di I/ O,
- un **regolatore di tensione**
- un'interfaccia USB che permette la comunicazione con il computer e l'esterno.

A questo hardware viene affiancato un ambiente di sviluppo integrato (IDE) multipiattaforma (per Linux, Apple Macintosh e Windows).

Questo software permette di scrivere programmi con un linguaggio semplice e intuitivo derivato da C e C++ liberamente scaricabile e modificabile.

I programmi in Arduino vengono chiamati **sketch**.

Arduino può essere utilizzato per lo sviluppo di oggetti interattivi stand-alone e può anche interagire, tramite collegamento, con software residenti su computer, come Adobe Flash, Processing,

La piattaforma hardware Arduino è spesso distribuita 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**.

In questo modo, chi lo desidera 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.

Questa possibilità ha consentito lo sviluppo di prodotti Arduino compatibili da parte di piccole e medie aziende in tutto il mondo: è quindi divenuto possibile scegliere tra un'enorme quantità di schede Arduino-compatibili.

Ciò che accomuna questi prodotti inerenti elettronica sperimentale e sviluppo è:

- il **codice sorgente** per l'ambiente di sviluppo integrato
- 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, sensori, display o attuatori**.

Dopo anni di sperimentazione, è oggi possibile fruire di un database di informazioni vastissimo.

L'hardware originale Arduino è interamente realizzato in Italia dalla Smart Projects, mentre i cloni della scheda possono essere realizzati da chiunque in qualsiasi parte del mondo.

Una scheda Arduino tipica consiste in un microcontrollore 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.

- **La programmazione del microcontroller**

In alcuni casi il microcontroller della scheda 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 una 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: la versione Uno utilizza un microcontrollore Atmega8U2 programmato come convertitore USB-seriale.

- **Funzionalità di input/output**

Per implementare il comportamento interattivo, Arduino è fornita di funzionalità di input/output (I/O), grazie alle quali essa riceve i segnali raccolti da sensori esterni.

Il comportamento della scheda è gestito dal microcontroller in base ai valori provenienti dai sensori e alle decisioni determinate dal particolare programma in esecuzione in quel momento sulla scheda.

L'interazione con l'esterno avviene attraverso attuatori pilotati dal programma per mezzo dei canali di output in dotazione.

- **Connettori di 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".

- **I/O digitale su Arduino UNO**

La Arduino Uno offre **14 connettori per l'I/O digitale (numerati da 0 a 13)**.

Non usare i pin 0 ed 1 (di solito usati per scopi speciali).

La direzione o modo di funzionamento, input o output, è decisa dallo *sketch* programmato sull'IDE.

6 dei 14 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 motore elettrico.

L'hardware di 6 dei pin di I/O (3,5,6, 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

La funzione accetta due parametri

- il primo dei quali è il pin pilotato
- 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

- **I/O analogico**

Sempre sulla *Uno*, sono presenti altri 6 connettori specificamente dedicati a ingressi di segnali analogici (collegati quindi ad una ADC), 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.

- **Caratteristiche di Arduino UNO R3:**

Microcontrollore:	ATmega328P
Flash memory:	32 KB
EEPROM	1 KB
SRAM	2 KB
pin I/O digitale	14 (0-13)
di cui con PWM	6 (3,5,6,9,10,11)
pin I analogico	6 (0-5)
Tipo USB	ATmega8U2
dimensioni	68,6 x 53,3 mm

- **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

- **Schede Arduino compatibili**

L'enorme quantità e l'estrema variabilità d'uso e di componenti rendono difficile definire univocamente una scheda Arduino-compatibile.

Solitamente, essa contiene un microcontroller a 8 16 o 32 bit AVR, PIC o ARM, con clock variabile tra 1 e 96 MHz. Molte schede incorporano componenti aggiuntivi pensati per i più svariati utilizzi.

- **Software**

L'ambiente di sviluppo integrato (IDE) di Arduino è un'applicazione multiplatforma 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.

Insieme con l'IDE vengono scaricati vari *sketch* di esempio, per introdurre l'utente alla programmazione della macchina; le tematiche sono molto basiche, come ad esempio gestire gli ingressi analogici e digitali, far accendere un led in modo pulsante e variabile; si possono però anche affrontare problemi più complessi come la gestione di un display LCD o di una scheda telefonica GSM

Oltre alle varie librerie già incorporate (più di una decina), l'utente può aggiungerne di nuove con uno strumento di importazione compreso nell'IDE.

Per vedere i risultati di uno sketch è attivabile dall'IDE una *finestra seriale* di monitoring, sulla quale far comparire l'output di istruzioni `Serial.print(parametro)` incorporate nello sketch stesso.