



ISTITUTO TECNICO INDUSTRIALE STATALE
G. GALILEI DI SAN SECONDO



SISTEMI EMBEDDED

Il caso Arduino



SISTEMA EMBEDDED

In elettronica e informatica, con il termine **sistema embedded** (generalmente tradotto in italiano con *sistema immerso o incorporato*) s'identificano genericamente tutti quei sistemi elettronici di elaborazione a microprocessore progettati appositamente per una determinata applicazione (*special purpose*) ovvero non riprogrammabili dall'utente per altri scopi, spesso con una piattaforma hardware *ad hoc*, integrati nel sistema che controllano ed in grado di gestirne tutte o parte delle funzionalità richieste.



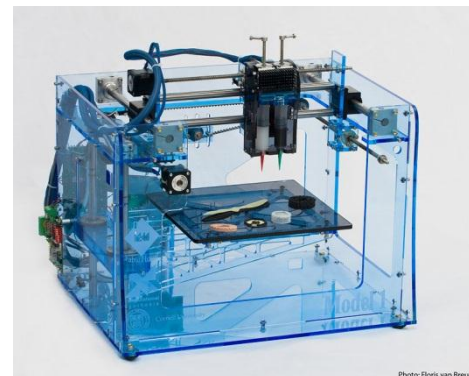
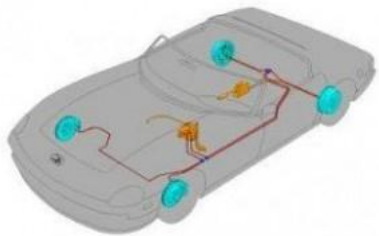


QUANTI MICROCONTROLLORI AVETE USATO OGGI?

Un microcontrollore è il «genere di computer in miniatura» che potete trovare in ogni genere di oggetti.

Se possiede dei tasti e un display, è **probabile che abbia anche un cervello a microcontrollore.**

- *Sistemi di frenata per automobili*
- *Macchine industriali*
- *Auto-focus delle fotocamere*
- *Telefoni cordless*
- *Card reader*
- *Sistemi per test medici*
- *Caricatori per batteria*
- *DVD player*
- *Navigatori*
- *Robot*
- *Impianti domotici*
- ...



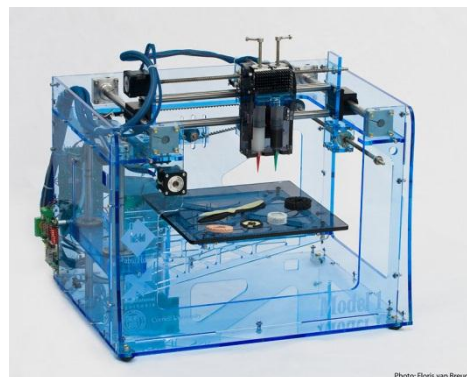
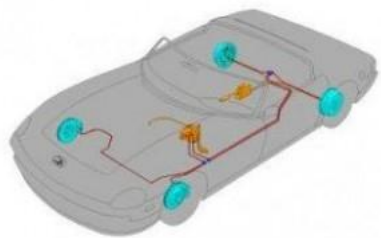


QUANTI MICROCONTROLLORI AVETE USATO OGGI?

I SE sono sistemi di elaborazione in stretta relazione con l'ambiente in cui operano. Essi sono diversi dai sistemi general purpose (PC).

Ognuno di noi viene a contatto con circa **100 dispositivi a uP embedded al giorno.**

- *Sistemi di frenata per automobili*
- *Macchine industriali*
- *Auto-focus delle fotocamere*
- *Telefoni cordless*
- *Card reader*
- *Sistemi per test medici*
- *Caricatori per batteria*
- *DVD player*
- *Navigatori*
- *Robot*
- *Impianti domotici*
- ...





MA NON POSSO USARE UN SEMPLICE PC?

La differenza sostanziale tra un PC ***general purpose*** e un SE è che il primo deve essere prima di tutto versatile, adatto ad ogni esigenza ed uso differente, mentre il secondo deve invece essere ottimizzato per un determinato impiego.

Un progetto realizzato con un Sistema Embedded dipende da:

1. Scenari di utilizzo;
2. Requisiti funzionali;
3. Vincoli di progetto

Nello spazio delle soluzioni, in base ai vincoli preposti si cercherà quella più adatta.



SCENARIO

Ambiente domestico



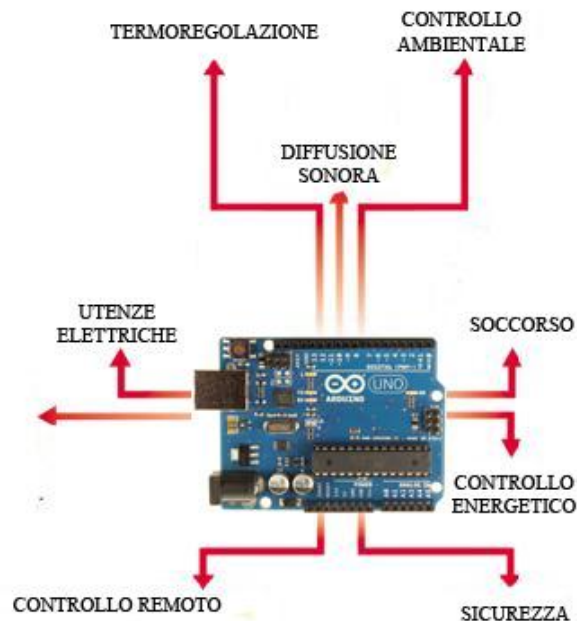


REQUISITI FUNZIONALI

Ambiente domestico



Interazione uomo-ambiente





VINCOLI PROGETTUALI

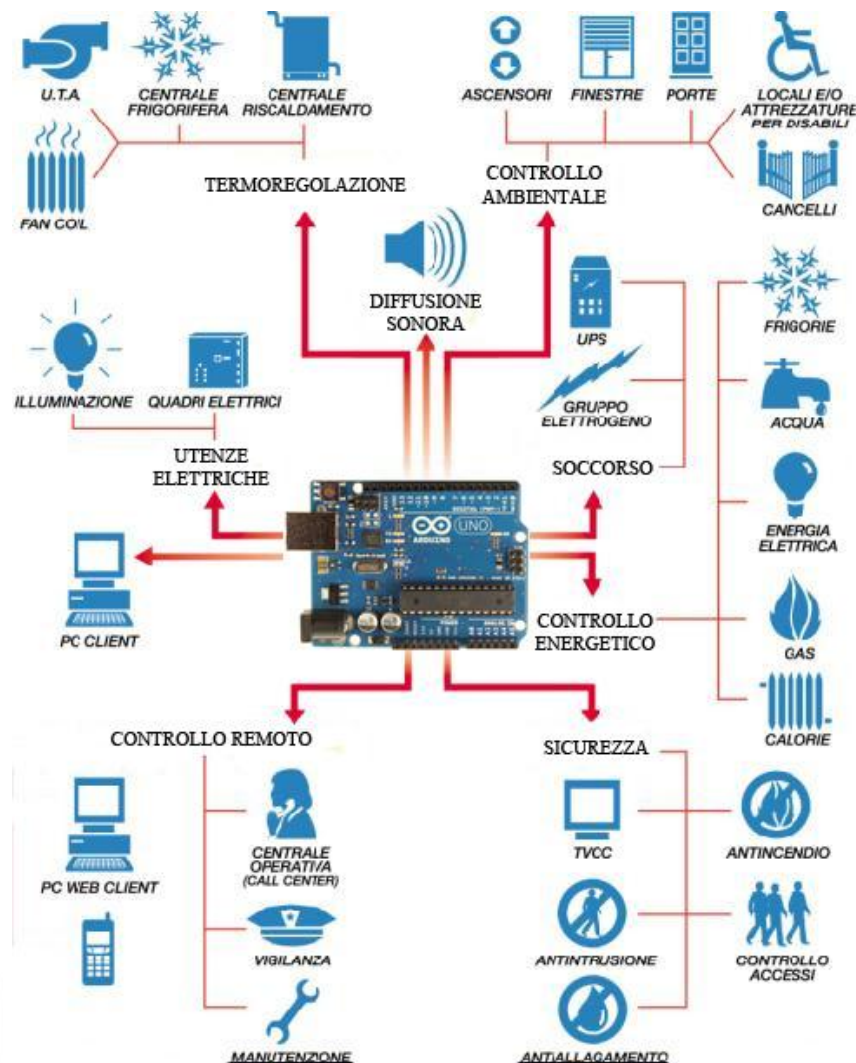
Ambiente domestico



Interazione uomo-ambiente



Integrazione di sensori e dispositivi
per rendere l'ambiente
INTELLIGENTE





UN ALTRO PUNTO DI VISTA



UN ALTRO PUNTO DI VISTA

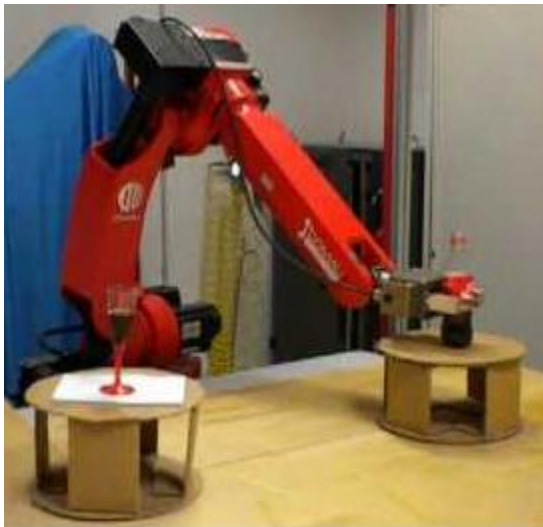
- Prodotto consumer (*la banale macchina fotografica!!!*)
- Oggetto da integrare in uno smartphone (*esempio per leggere i codici QR*)
- Funzione FOTO in una video camera (*per ampliare funzionalità di un oggetto*)
- Rilevatore di calore (*per scopi militari*)
- Dispositivo di inquadramento per un robot (*per scopi assistenziali*)
- Oggetto da integrare in giocattoli (esempio la Kinect della XBOX)
- ...





CONTESTI APPLICATIVI PER UN SISTEMA EMBEDDED

- **Infrastrutture ed edilizia** (autostrade, ponti, aeroporti, uffici, abitazioni, commerciale, ...)
- **Sistemi mobili** (persone, animali, salute, ...)
- **Industriale** (automobili, aerospaziale, automazione, consumer, ...)





I PRINCIPALI SISTEMI EMBEDDED

ARDUINO

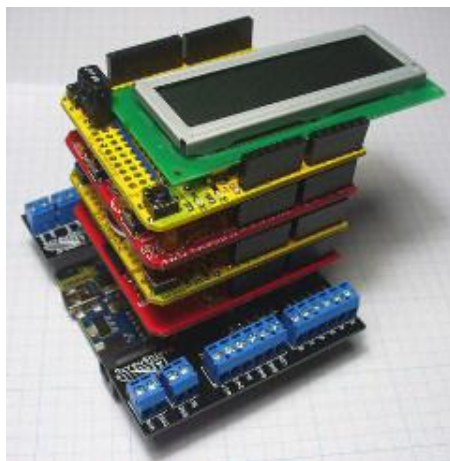
RASPBERRY

PIC

FRIENDLY ARM

OPEN PICUS

LIBELIUM



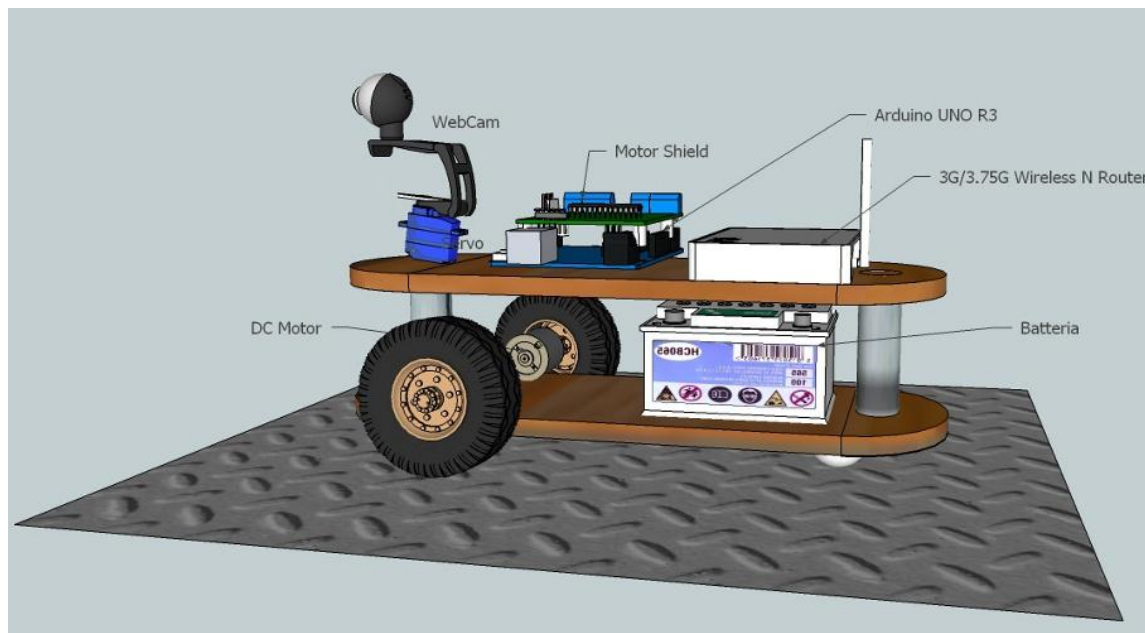


LA FILOSOFIA ARDUINO

La **prototipizzazione** è il cuore della filosofia Arduino: facciamo cose e costruiamo oggetti che interagiscono con altri oggetti, persone e reti. Ci sforziamo di trovare un modo più facile e veloce di prototipare nel modo più economico possibile.

L'obiettivo per molti principianti che si avvicinano all'elettronica è quello di vedere subito qualcosa che funziona, in modo da sentirsi motivati ad eseguire il passo successivo o motivare qualcun altro a fornirci il denaro per farlo.

Perché perdere tempo ed energia a costruire qualcosa da zero quando qualcuno ha già realizzato dispositivi già pronti all'uso?





LA FILOSOFIA ARDUINO

Riutilizzare la tecnologia esistente è una delle forme migliori di **tinkering**. Procurarsi giocattoli economici o vecchie attrezzature che andrebbero buttate e modificarle per creare qualcosa di nuovo è uno dei modi migliori per ottenere ottimi risultati.

Con il termine **patching** si intende la possibilità di costruire sistemi complessi collegando tra loro dispositivi semplici. Esempio i musicisti hanno costruito suoni, cercando infinite combinazioni mettendo insieme (patching) diversi moduli con cavi.





LA FILOSOFIA ARDUINO

“Scusate l’espressione, ma gli orinatori maschili sono più intelligenti dei computer. I computer sono isolati da ciò che li circonda” (A. Pentland – MIT Lab, 1998)



Si possono implementare nuovi modi di interagire con il software, sostituendo i tasti con dispositivi in grado di sentire l’ambiente.

Se smontate una tastiera di un computer scoprirete un dispositivo molto semplice (ed economico). Il suo cuore è una piccola scheda con una serie di contatti che vanno ai due strati di plastica che contengono le connessioni dei tasti.

Se provate a fare ponte tra i due contatti, sullo schermo apparirà la lettera corrispondente. Se ora immaginate di attaccare un sensore di movimento e lo collegate alla tastiera, noterete che ogni volta che passa qualcuno, sul computer verrà visualizzata una lettera. Se riuscite a creare un software che “correli” questi dati **avete reso il computer intelligente più di un orinatoio.**



LA FILOSOFIA ARDUINO

Amiamo i rifiuti ed i giocattoli!!

Oggigiorno la gente butta moltissimi oggetti tecnologici: vecchie stampanti, computer, strane macchine da ufficio, equipaggiamenti tecnici ma anche militari.

Di questa tecnologia di scarto è sempre esistito un fiorente mercato, specie tra gli hacker più poveri o quelli che sono agli inizi.

Ivrea era la città sede della Olivetti. A metà degli anni '90 buttava via tutti i suoi rifiuti dal rottamaio della zona. Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, e David Mellis hanno passato innumerevoli ore a cercare aggeggi di ogni sorta a prezzi bassissimi per riutilizzarli per i loro prototipi.

Nel 2005 infatti nasce Arduino sempre ad Ivrea.





LA FILOSOFIA ARDUINO

Collaborazione

La collaborazione tra utenti è uno dei principi chiave nel mondo di Arduino.

Attraverso il forum su www.arduino.cc/it, persone di diverse parti del mondo si aiutano reciprocamente a imparare ad usare la scheda e i dispositivi con cui poterla integrare. Il team di Arduino incoraggia le persone a collaborare a livello locale, oltre ad aiutare a fondare gruppi utenti nelle città che visita.

Esiste anche un Wiki di nome “playground” dove gli utenti documentano le loro scoperte.

La cultura della condivisione e dell’aiuto reciproco è un vanto per Arduino.

[Main Site](#) [Blog](#) [Playground](#) [Forum](#) [Labs](#) [Store](#)





ARDUINO: creare è un gioco da ragazzi

[Arduino: creare è un gioco da ragazzi](#)



COME FUNZIONA UN MICROCONTROLLORE

Per quanto potente, il **microprocessore** integra sul chip solo la logica di elaborazione mentre richiede sempre delle unità esterne - memorie, gestori di segnali e dispositivi periferici per scambiare informazioni e interagire con l'esterno.

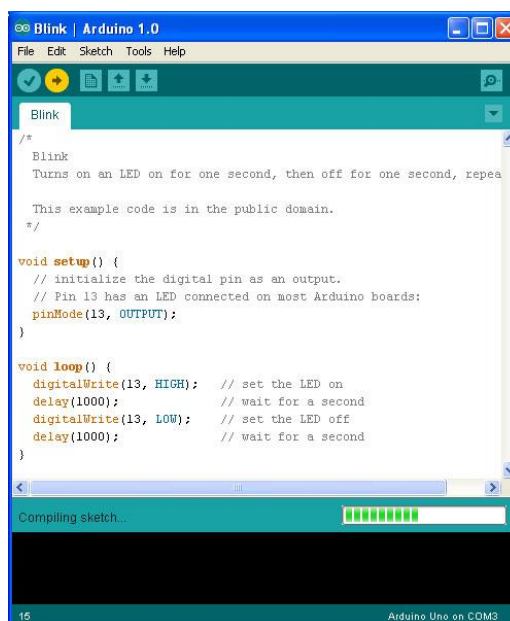
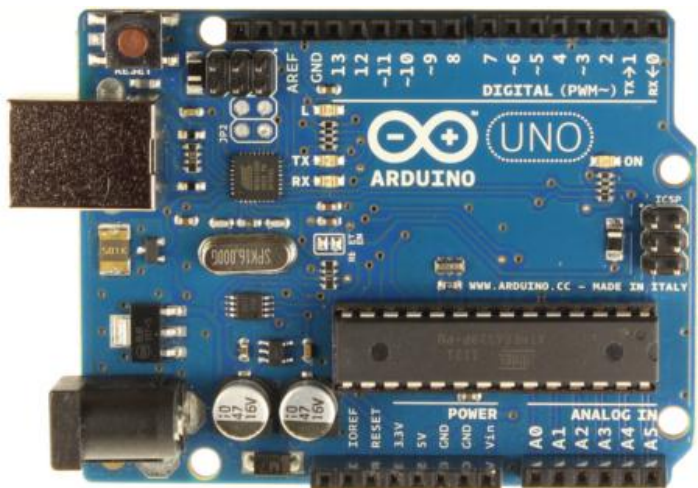
Il **microcontrollore** è invece un sistema completo, che integra in uno stesso chip il processore, la memoria permanente, la memoria volatile e i canali (pin) di I/O, oltre ad eventuali altri blocchi specializzati.

CONFRONTO TRA MICROPROCESSORE E MICROCONTROLLORE		
Caratteristica	Microcontrollore	Microprocessore
Velocità massima di clock	200 MHz	4 GHz
Capacità elaborativa in MegaFLOPS	200	5 000
Potenza minima dissipata in watt (in stato di elaborazione)	0,001	50
Prezzo minimo per singola unità in USD	0,5	50
Numero di pezzi venduti annualmente (in milioni)	11 000	1 000



ARDUINO – LE PARTI PRINCIPALI

1. Una **scheda elettronica** (con le porte di I/O e un microcontrollore contenente un firmware)
2. Un **ambiente di programmazione** (ambiente di sviluppo sketch, linguaggio Arduino, Wiring ed eventualmente Processing)
3. La **comunità degli utenti**.



[Main Site](#) [Blog](#) [Playground](#) [Forum](#) [Labs](#) [Store](#)



🏠 | Arduino Forum

Using Arduino

- +** **Installation & Troubleshooting**
For problems with Arduino itself, NOT your project
Last post: [Re: Problem installing A...](#) by [Louis Davis](#) on [Today](#) at 06:06:00
- +** **Project Guidance**
Advice on general approaches or feasibility
Last post: [Re: Arcade button connec...](#) by [Osgeld](#) on [Today](#) at 06:31:49 PM
- +** **Programming Questions**
Understanding the language, error messages, etc.
Last post: [Re: running switch case ...](#) by [PaulS](#) on [Today](#) at 06:31:44 PM
- +** **General Electronics**
Resistors, capacitors, breadboards, soldering, etc.
Last post: [Re: Transistor issue](#) by [dc42](#) on [Today](#) at 05:49:58 PM
- +** **Microcontrollers**
Standalone or alternative microcontrollers, in-system programming
Last post: [Re: How do I use Arduino...](#) by [simplex](#) on [Today](#) at 05:34:00
- +** **LEDs and Multiplexing**
Controlling lots of inputs and outputs
Last post: [Re: RGB SMT LED Cube, re...](#) by [Hippynerd](#) on [Today](#) at 06:24:00



IL FIRMWARE

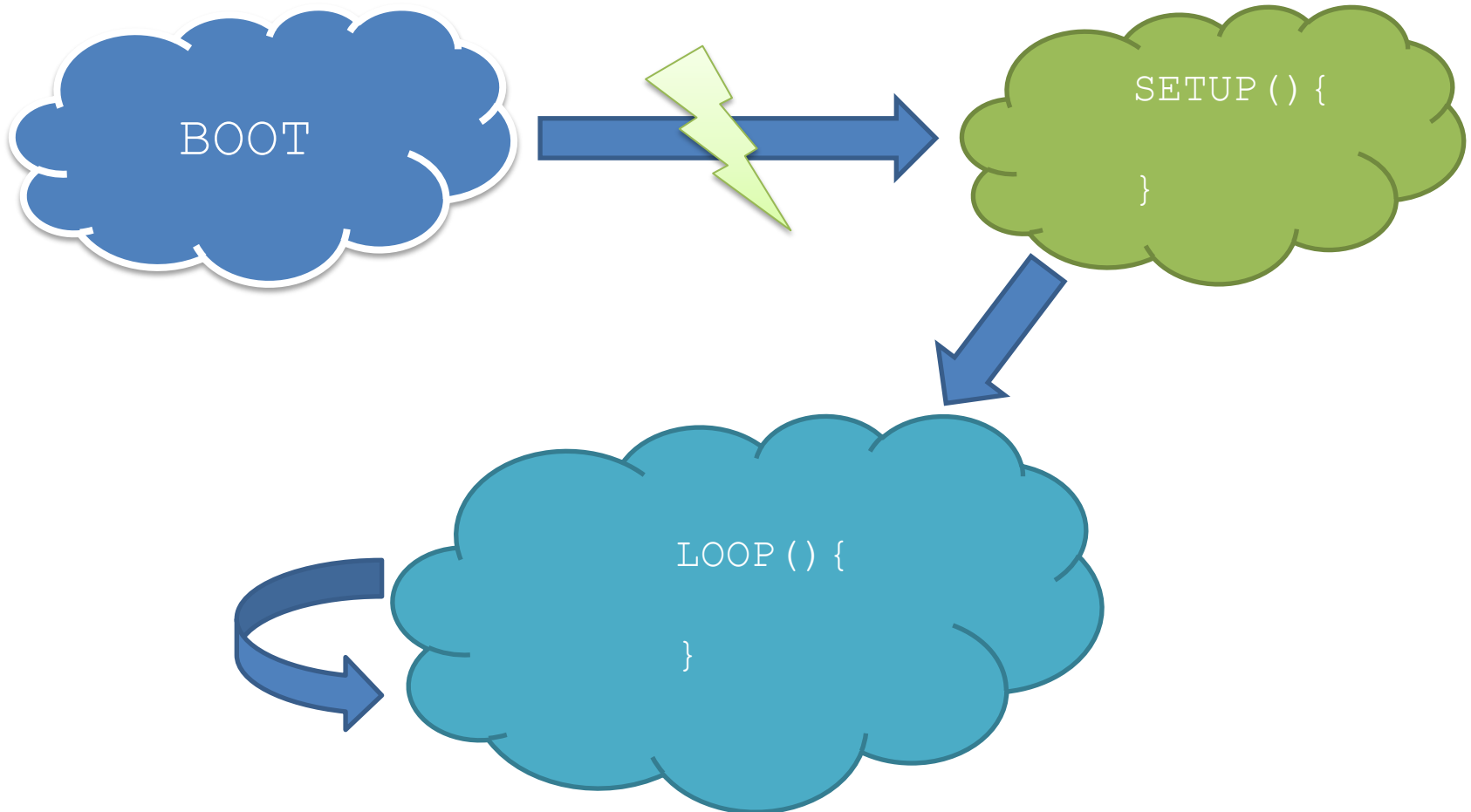
Il **firmware** è un programma, inteso come sequenza d'istruzioni, integrato direttamente in un componente elettronico nel senso più vasto del termine (integrati, schede elettroniche, periferiche). Lo scopo del programma è di *avviare* il componente stesso e consentirgli di interagire con altri componenti tramite l'implementazione di protocolli di comunicazione o interfacce di programmazione.

Il termine deriva dall'unione di "**firm**" (azienda) e "**ware**" (componente), indica che il programma non è immediatamente modificabile dall'utente finale, ovvero risiede stabilmente nell'hardware integrato in esso, e che si tratta del punto di incontro fra componenti logiche e fisiche, ossia fra hardware e software.

Quando si parla di firmware per una scheda elettronica (come una scheda di espansione per computer) questo generalmente trova posto all'interno di una memoria ROM o flash.



ARDUINO - LO SKETCH





ARDUINO – IL MICROCONTROLLORE

Arduino adotta i microcontrollori Atmel AVRmega.

Arduino Uno ha un'interfaccia USB (del tipo 'stampante') per il collegamento e la programmazione da PC.

E' anche possibile collegare un programmatore esterno per AVR, attraverso l'interfaccia e il connettore ISP. Tuttavia, per la presenza di un bootloader precaricato, il programmatore esterno non è indispensabile.

Inoltre, si può usare il linguaggio di programmazione 'Processing', sviluppato al MIT e basato su Java. Ovviamente, **'Processing' da solo non è Arduino.**



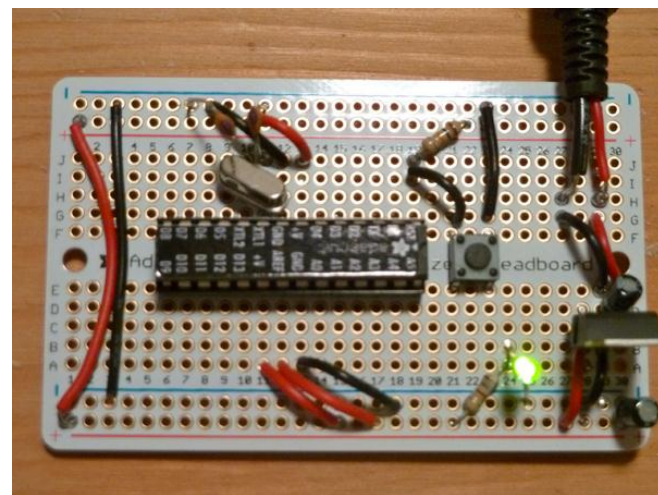
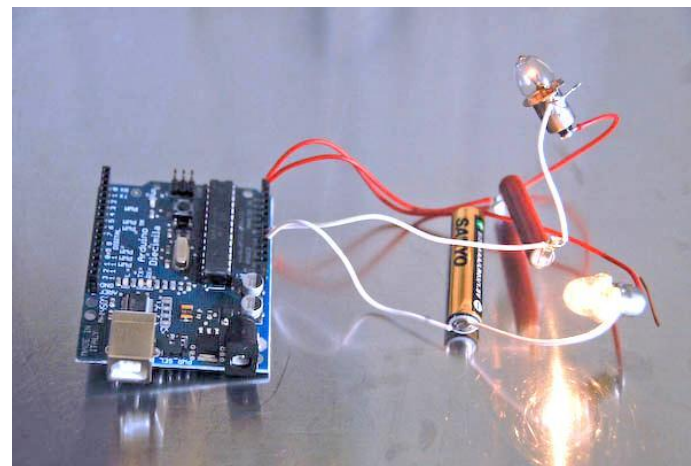


ARDUINO – COME USARLO

Arduino, una volta programmato, può funzionare autonomamente, **alimentato da una batteria, da un alimentatore esterno oppure collegato a un PC e alimentato dalla USB.**

Basata su una semplice scheda input/output (I/O) e un ambiente di sviluppo che implementa il linguaggio processing (dialetto di C). Arduino può essere usato per sviluppare oggetti interattivi indipendenti o può essere collegato a un software sul computer (come Flash, processing, Java, ...).

Le schede possono essere assemblate a mano o comprate preassemblate; l'IDE è scaricabile da www.arduino.cc





ARDUINO – LA SEMPLICITÀ NEL PROGRAMMARE

Strutture generali

setup()

loop()

Strutture di controllo

if

if .. else

for

switch case

while

do .. while

break

continue

return

goto

Segni di interpunzione usati

;

{ }

//

/*/

Direttive

#define

#include

Tipi di dati

void

boolean

char

insigne char

byte

int

insigne int

word

long *intero a 32 bit con segno*

unsigned long *intero a 32 sempre positivo*

float

double

string (di caratteri)

String (oggetto)

Array



ARDUINO – LA SEMPLICITÀ NEL PROGRAMMARE

I/O Digitali

pinMode()
digitalWrite()
digitalRead()

I/O Analogici

analogReference()
analogRead()
analogWrite()

Avanzate

tone()
noTone()
shiftOut()
pulseIn()

Gestione del tempo

millis()
micros()
delay()
delayMicroseconds()

Funzioni matematiche

min(), max(), abs()
constrain() *costringe un valore in un intervallo a, b*
operazioni + - * / %
map() *trasforma un intero da un intervallo ad un altro*
pow() *Potenza*
sqrt() *radice quadrata*
sin(), cos(), tan()

Numeri casuali

randomSeed()
random()

Operazioni sui bit e i bytes

lowByte (), highByte()
bitRead(), bitWrite()
bitSet(), bitClear()
bit()
&(and su bit)
|(or su bit)
^(xor)
~(not su bit)
<<(shift a sinistra)
>>(shift a destra)



ARDUINO – LA SEMPLICITÀ NEL PROGRAMMARE

Costanti predefinite

LOW, HIGH

INPUT, OUTPUT

Utilità

`sizeof()` *fornisce le dimensioni in byte di una variabile*

Gestioni interrupt esterni

`attachInterrupt()`

`noInterrupt()`

Interrupt Interni

`interrupts()`

`noInterrupts()`

Comunicazione seriale

`Serial`

Confronti

`=` `!=` (diverso) `<` `>` `<=` `>=`

Booleani

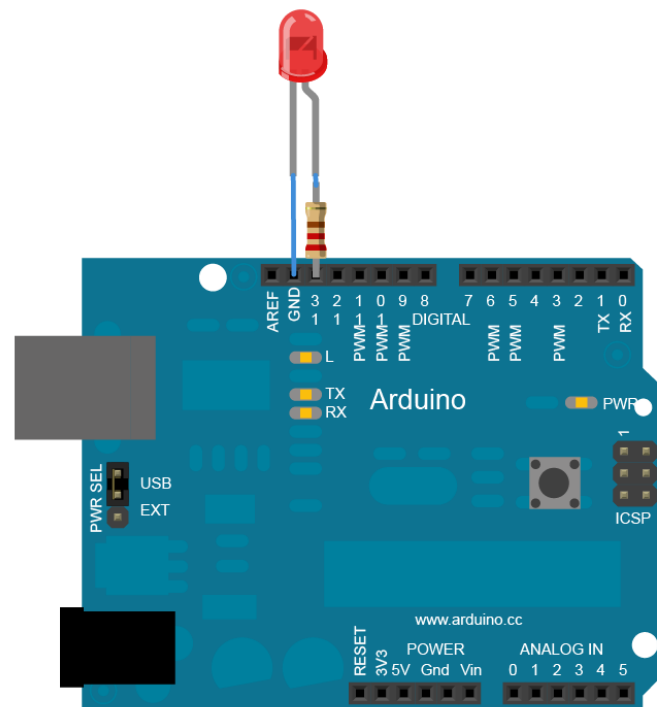
`&&` (and) `||` (or) `!` (not)



ARDUINO – LA SEMPLICITÀ NEL PROGRAMMARE

Facciamo “Blinkare” un LED.

```
// Pin 13 has an LED connected on most Arduino boards.  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    digitalWrite(led, HIGH);    // turn the LED on  
    delay(1000);               // wait for a second  
    digitalWrite(led, LOW);    // turn the LED off  
    delay(1000);               // wait for a second  
}
```





ARDUINO – LA SEMPLICITÀ NEL PROGRAMMARE

Facciamo “Blinkare” un LED, ma in modo più intelligente.

```
const int ledPin = 13;
int ledState = LOW;
long previousMillis = 0; // memorizza il tempo dell'ultimo aggiornamento
                          // long perchè il tempo è in millisecondi
long interval = 1000;    // interval at which to blink (milliseconds)

void setup() {
    pinMode(ledPin, OUTPUT); // metto il piedino in Output
}

void loop() {
    unsigned long currentMillis = millis();
    if(currentMillis - previousMillis > interval) {
        // save the last time you blinked the LED
        previousMillis = currentMillis;
        // se il LED è On mettilo OFF e viceversa:
        if (ledState == LOW)
            ledState = HIGH;
        else
            ledState = LOW;
        digitalWrite(ledPin, ledState);
    }
}
```



ARDUINO – LE CARATTERISTICHE

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz





ARDUINO – LE COMMUNITY

<http://arduino.cc/forum/>

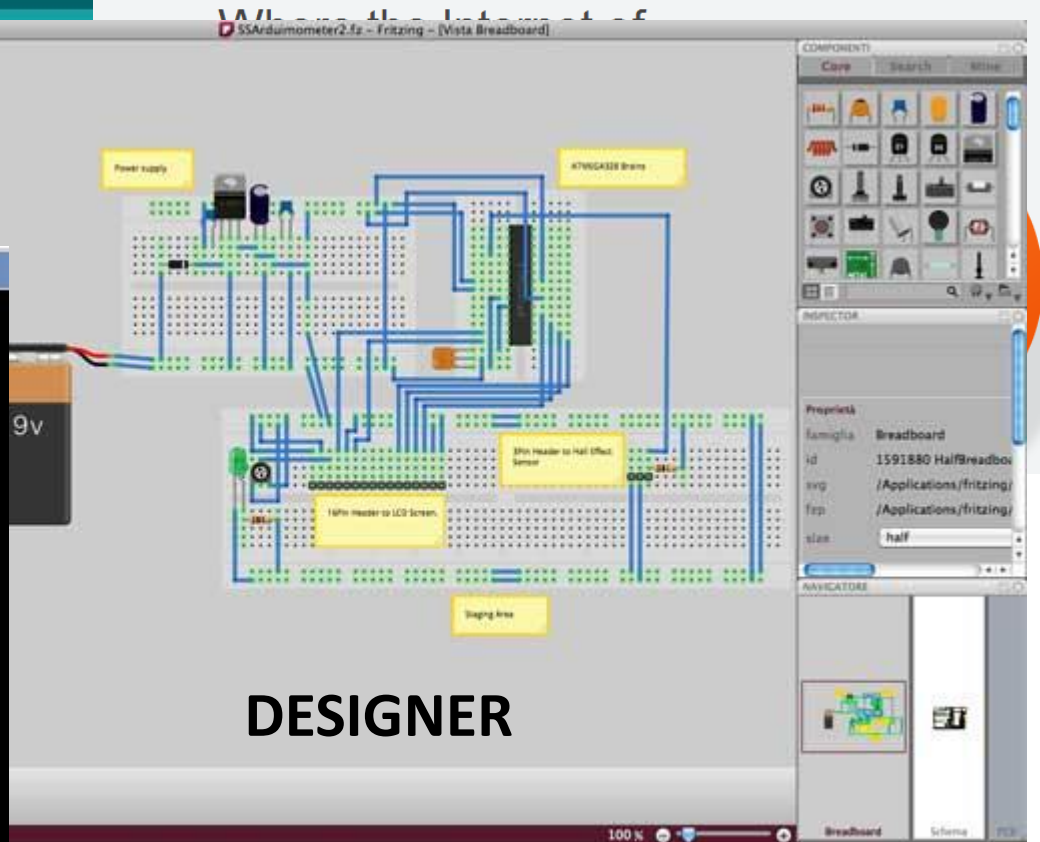
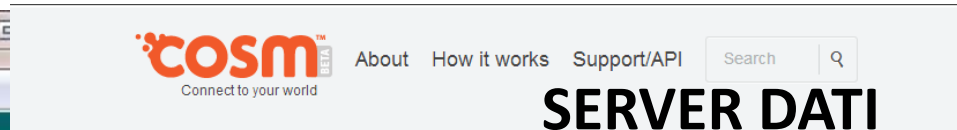
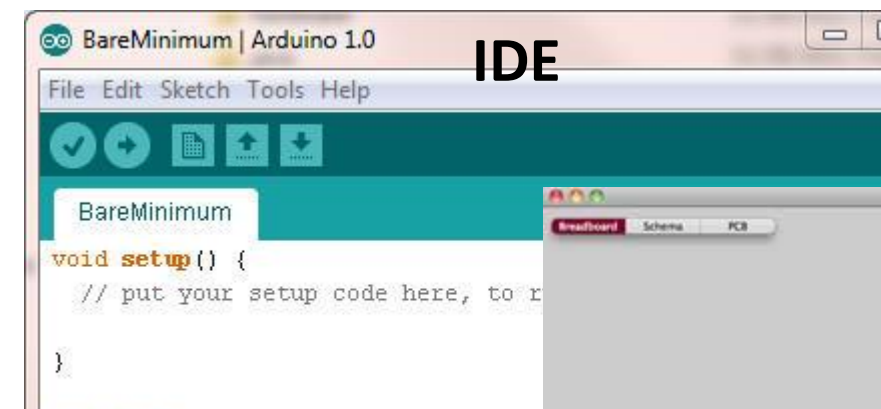
<http://www.instructables.com/>

www.google.it

<http://it.emcelettronica.com/arduino>



ARDUINO – GLI STRUMENTI PER INIZIARE





DIY – DO IT YOURSELF

(<http://www.doityourself.com/>)

Do-It-Yourself (DIY) è un termine usato per descrivere la progettazione, modifica o riparazione di qualcosa senza l'ausilio di esperti o professionisti. La frase fai da te è entrata in uso comune nel 1950 come riferimento ai progetti di miglioramento domestico che le persone potrebbero autonomamente fare indipendentemente.

Negli ultimi anni, il termine DIY ha assunto un significato più ampio che copre una vasta gamma di abilità. Grazie alla concezione di fai da te sono nate vere e proprie comunità di nicchia nelle quali è possibile discutere, commentare, migliorare progetti con dispositivi differenti (come nel caso di Arduino). Per questo motivo, non sono più necessari mesi di studio per sviluppare progetti basati su questi dispositivi: bastano, infatti, poche settimane per fare progetti veramente interessanti.

Per esempio Sebastian Alegria, un ragazzo di quattordici anni cileno, ha creato un sistema di rilevamento per i terremoti e li pubblica su twitter ed ha anticipato un progetto governativo di un anno.



ISTITUTO TECNICO INDUSTRIALE STATALE
G. GALILEI DI SAN SECONDO



COSA SI FA CON ARDUINO



DSG Lab

<http://dsg.ce.unipr.it>

- peer-to-peer computing
- mobile and pervasive computing
- service-oriented architectures
- autonomic computing
- inter-vehicular networks
- context-awareness and localization



Distributed Systems Group

Dip. di Ingegneria dell'Informazione
Università degli Studi di Parma



About us

- DSG Home
- People
- Research
- Publications
- Students
- Visitors
- Blog

Projects

- DSG Projects
- European Projects
- Completed Projects

User login

Username: * Password: *

DSG

Welcome to the web pages of DSG, the Distributed Systems Group of the University of Parma located in the Department of Information Engineering (DII). We are specialized in modeling, performance evaluation and implementation of distributed systems and applications.

Research topics:

- peer-to-peer computing
- mobile and pervasive computing
- service-oriented architectures
- autonomic computing
- inter-vehicular networks
- context-awareness and localization

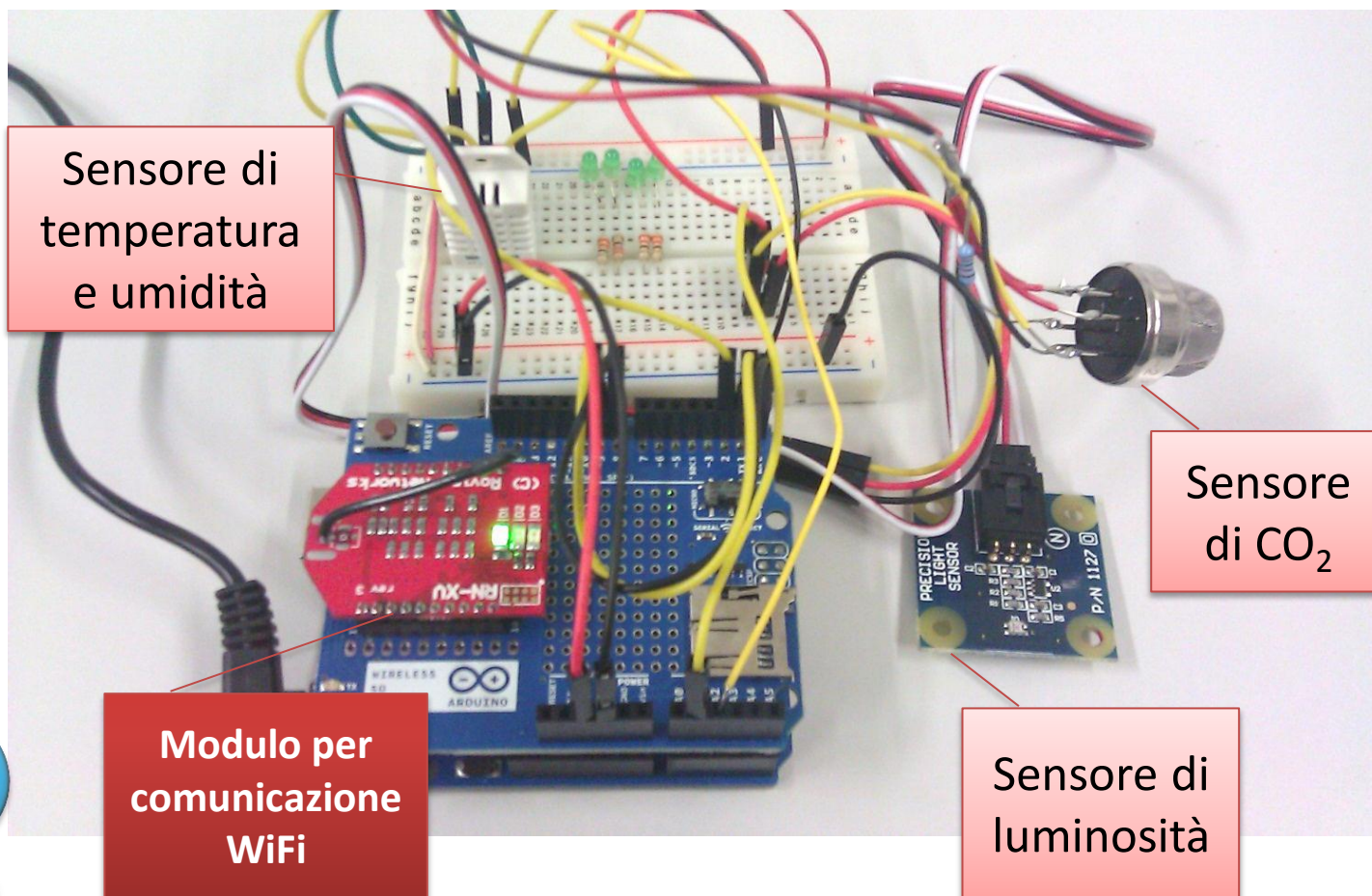
Current and Active Projects





Scheda per il monitoraggio ambientale

Scheda Arduino UNO che raccoglie i dati ambientali
e li invia al server attraverso un modulo WiFi.

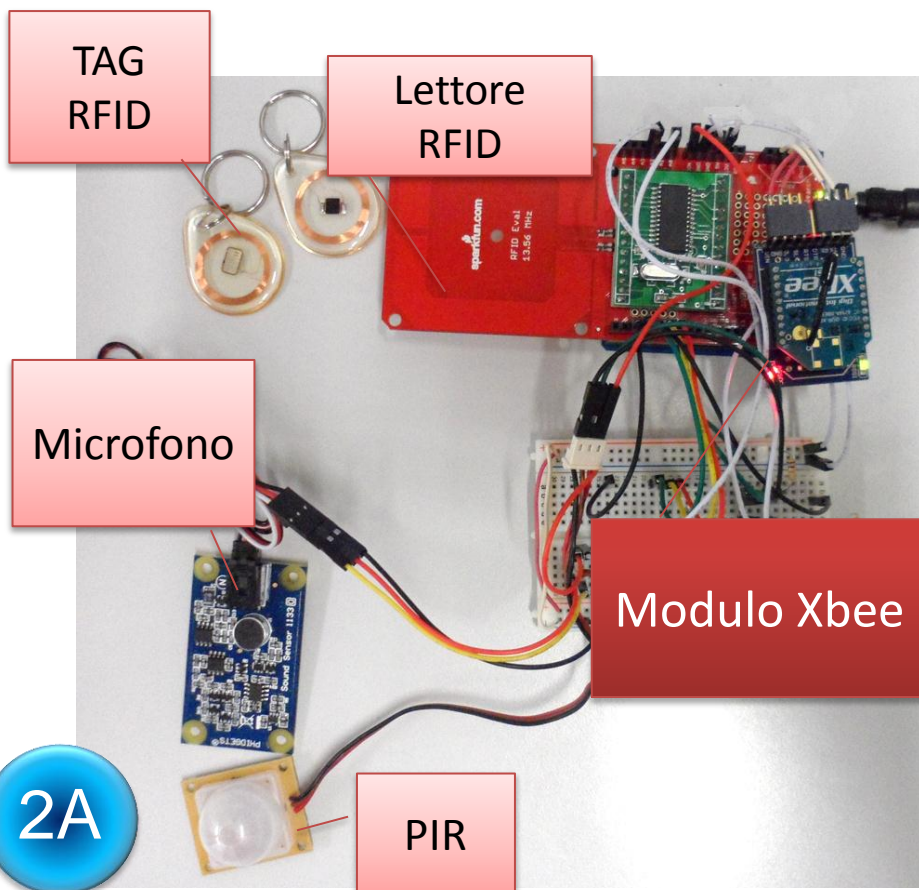


1

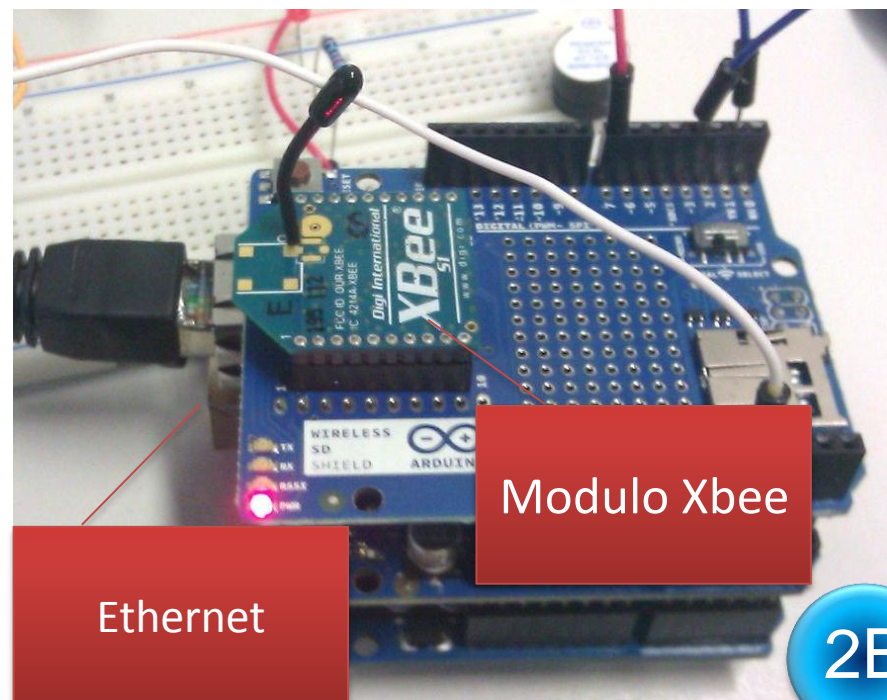


Schede per il riconoscimento e l'identificazione

Schede Arduino UNO che monitorano la presenza di persone all'interno dell'ambiente.
Comunicano tra di loro con il protocollo ZigBee e inviano i dati al server
attraverso la porta Ethernet.



2A



2B



Scheda per gli attuatori e l'Usb Host

Scheda Arduino Mega ADK con cui vengono simulati gli attuatori. È presente anche un Usb Host attraverso cui è possibile collegare un device Android.

Ventole = impianto di
condizionamento

Servo motore = Finestre

Led = Illuminazione

Ethernet

3



COSA SI FA CON ARDUINO

- Giochi:

Quadricottero (prima solo per scopi militari)

<http://www.scoutuav.com/category/videos/>

Controller “Casalingo”

<http://arduino.cc/en/Main/ArduinoBoardEsplora>



COSA SI FA CON ARDUINO

- Strumenti musicali:

Abiti musicali

<http://www.gadgetblog.it/post/28173/ruffletron-e-un-prototipo-di-abito-musicale-che-sindossa-con-arduino>

Controller MIDI

<http://www.coagula.org/content/it/pages/come-costruire-un-midi-ribbon-controller-con-arduino>

Arpa Laser

<http://www.youtube.com/watch?v=sLVXmsbVwUs>



COSA SI FA CON ARDUINO

- Ambito medico:
Dispositivo che analizza e rileva il DNA
<http://openpcr.org/>



COSA SI FA CON ARDUINO

- Ambito assistenziale:
Guanti che rilevano il linguaggio dei segni e li trasformano in suoni
<http://www.wikiculture.net/2012/05/15/quando-la-tecnologia-riesce-ad-abbattere-le-barriere/>



COSA SI FA CON ARDUINO

- Ambito artistico:

Txtbomber, è un dispositivo con il quale, inserendo un messaggio di testo come su un cellulare, è possibile poi aerografarlo dove si vuole con un semplice spruzzo di bomboletta

<http://arduino.cc/blog/2010/07/29/txtbomber-prints-messages-on-walls/>

Stampanti 3D

http://reprap.org/wiki/Main_Page



COSA SI FA CON ARDUINO

- Ambito naturalistico:
Dispositivo che rileva lo stato del benessere di una pianta e che interagisce con un account twitter
<http://comunitadigitali.blogosfere.it/2012/08/arduino-e-la-pianta-che-si-innaffia-da-sola.html>



COSA SI FA CON ARDUINO

- Ambito industriale:
Google si è interessata a questi dispositivi creando un Kit di sviluppo accessori (ADK) con il quale è possibile interagire con uno smartphone o tablet Android direttamente con una scheda Arduino
<http://www.youtube.com/watch?v=XEe3dYBj7RE>



PROGETTI DA REALIZZARE

- **Robottino**, una macchina taglia erba robot. Una **falciatrice robotizzata su base arduino**. Un filo perimetrale sotterrato recinta l'area di lavoro della falciatrice in modo che aiuole (ma soprattutto marciapiedi e strade) non vengano falciate :)
- Controllo strumenti musicali via **arduino, touch screen e pure-data**.
- **Macchina radio comandata via bluetooth** (possibilmente con **connessione diretta** quella che ho fatto io presuppone un po' di casino in più)
- **CNC con pezzi di vecchia stampante ink-jet**
- Sperimentazione di board **arduino clone cinese** dalle caratteristiche interessanti ma dalla documentazione assente (o aliena). La board in questione ha integrata la possibilità di **connessione ethernet, wifi, e socket per Xbee e forse fa anche il caffè**. Tutto integrato ad un prezzo + che interessante.



PROGETTI DA REALIZZARE

- Robot
- Impianto domotico
- Macchina RC ([link](#))
- "contro-campanella" della scuola
- Garduino ([link](#))
- Robot che segue pista nera
- Middleware per arduino



PROGETTI DA REALIZZARE

Progetti più piccoli

- Timer con allarme
- Impianto semaforico
- Strumento musicale (es con buzzer)
- Stazione metereologica
- Identificazione IR di un telecomando ([link](#))
- Seriale su display LCD
- Contagiri per bicicletta ([link](#))
- Lampada RGB ([link](#))
- Cubo di led ([link](#))
- <http://www.instructables.com/id/Intro-to-Arduino/>
- <http://www.instructables.com/id/Secret-Knock-Detecting-Door-Lock/>

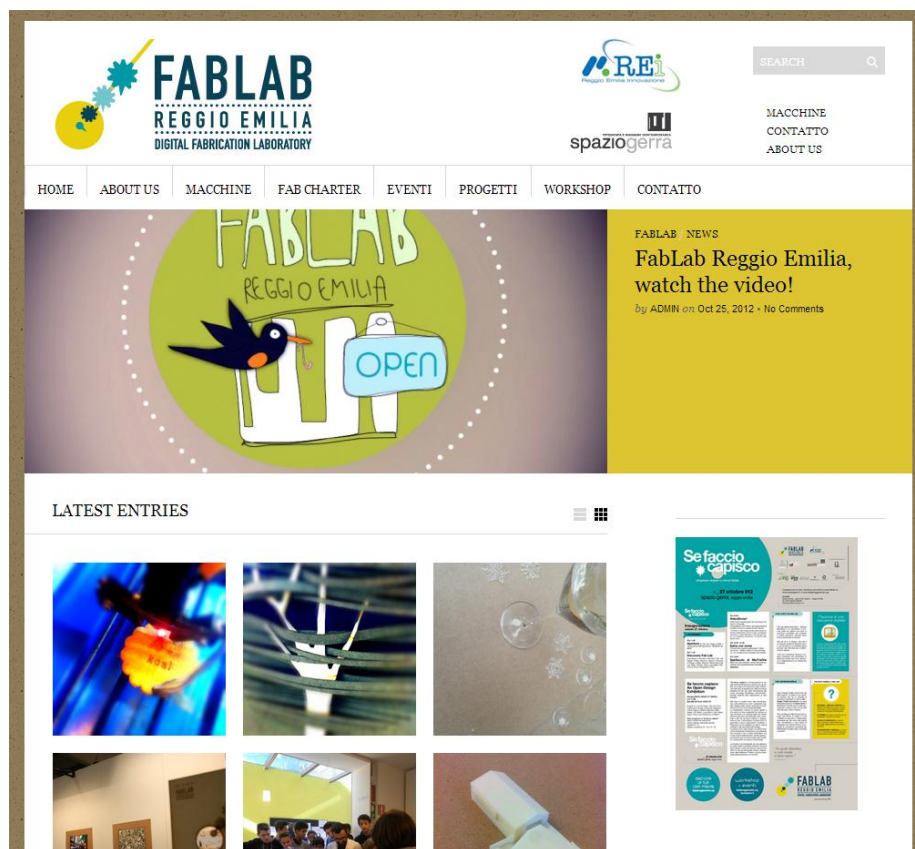


EVENTI

Arduino da Giugno 2012 organizza dei workshop per far conoscere la piattaforma e per condividere le idee di chiunque voglia partecipare.

Il prossimo workshop #arduinotour, si terrà a **Reggio Emilia a fine gennaio (25-26-27)**, presso il neonato [Fablab](#) ospitato all'interno dello [Spazio Gerra](#).

Se volete portare l'#arduinotour a casa vostra [riempite questo form](#). Se volete spargere il verbo fate il [like sulla pagina dell'arduinotour su facebook](#).



<http://www.fablabreggioemilia.org/>



ISTITUTO TECNICO INDUSTRIALE STATALE
G. GALILEI DI SAN SECONDO



RIFERIMENTI

www.mancio90.it

mirkomancin90@gmail.com

