

Laboratorio di Elettronica

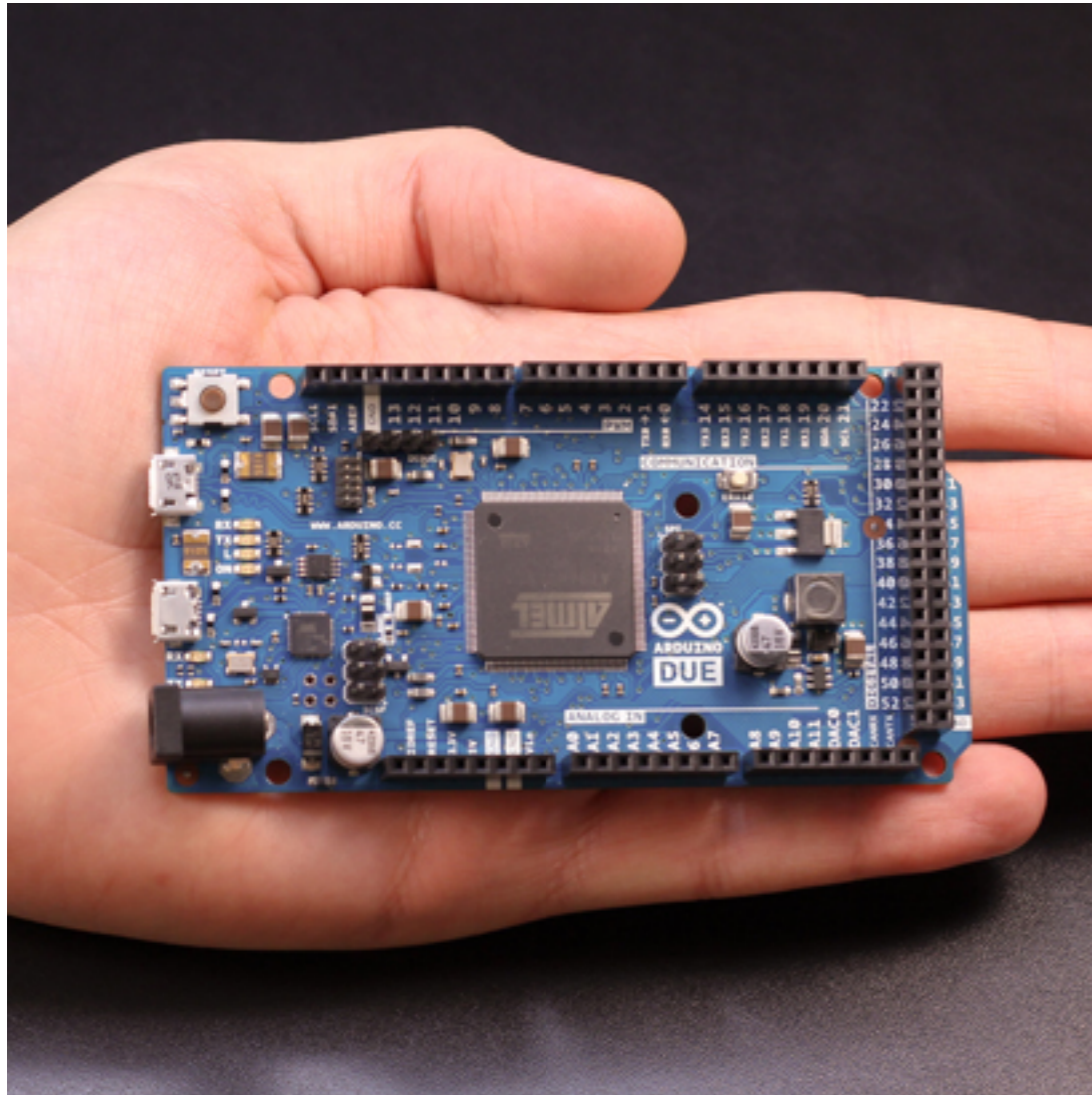
Arduino Lezione I

Descrizione della board e installazione

Lezione I

- **Arduino: introduzione ai componenti fondamentali**
- **Hardware: microcontrollore e I/O**
- **Software: come programmare il microcontrollore**
- **Installazione passo passo**
- **Esercizio (blinking LED)**

Cosa è Arduino



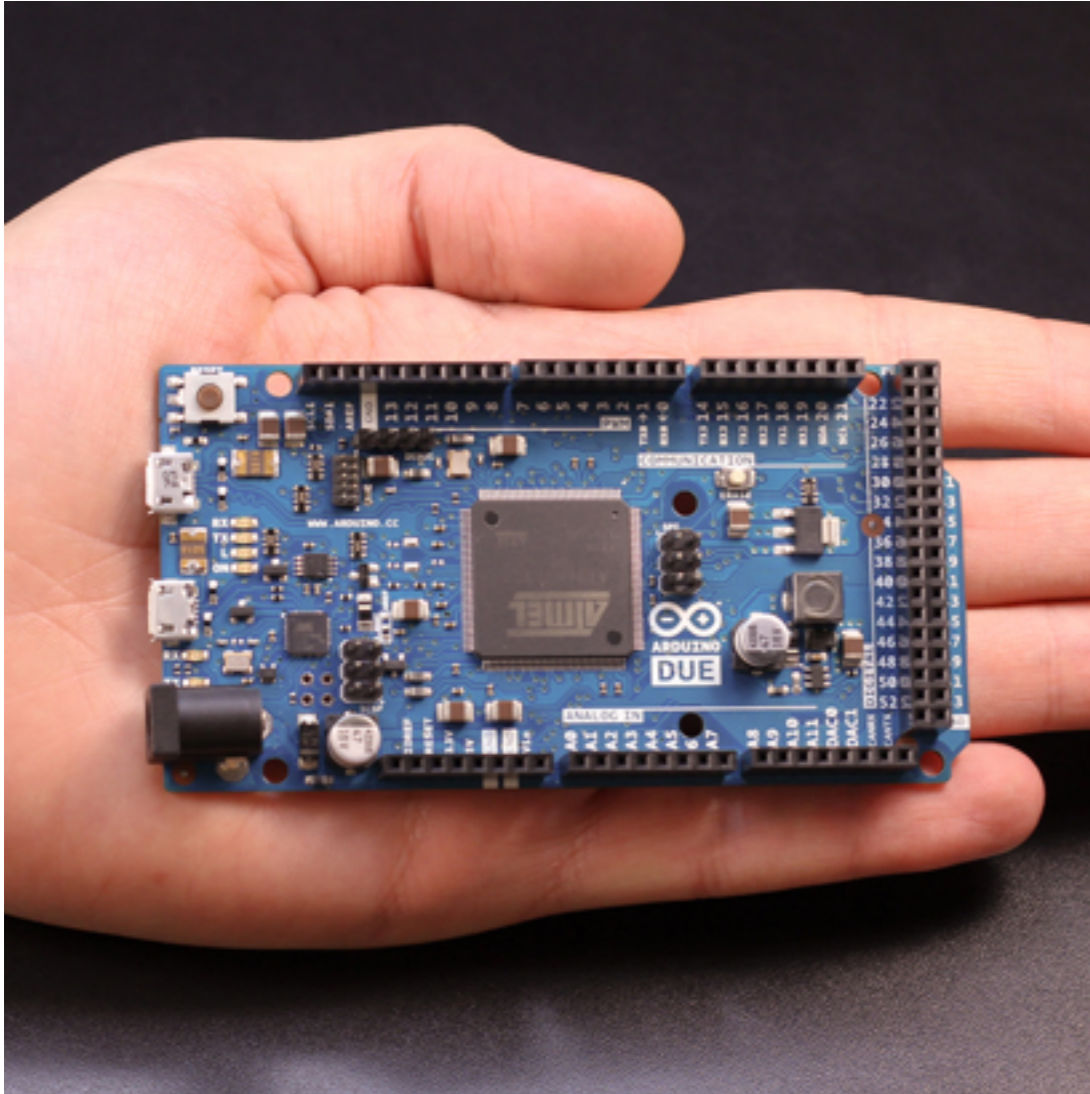
Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

da [Arduino.cc](https://www.arduino.cc)

Componenti principali:

- microcontrollore
- pin di I/O analogici e digitali
- memoria flash
- porta USB per comunicazione in maniera seriale

Cosa può fare Arduino



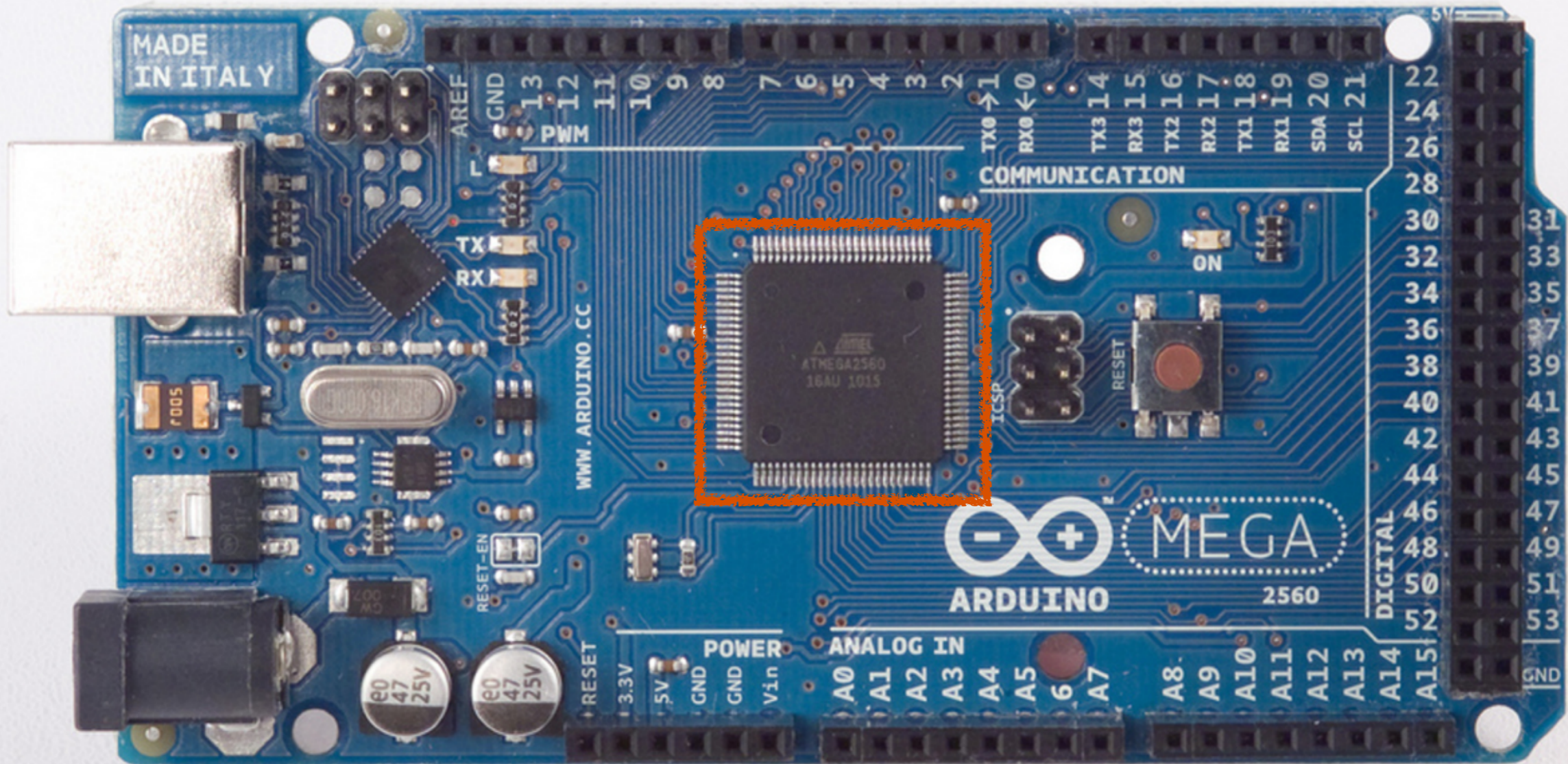
Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

da [Arduino.cc](https://www.arduino.cc)

- Leggere sensori
- Controllare periferiche
- Comunicare tramite porta seriale
- Programmabile

Microcontrollore

ATMega 2560



Microcontrollore

Evoluzione del microprocessore per applicazioni specifiche.
Ha una memoria interna ed è programmabile.

Componenti:

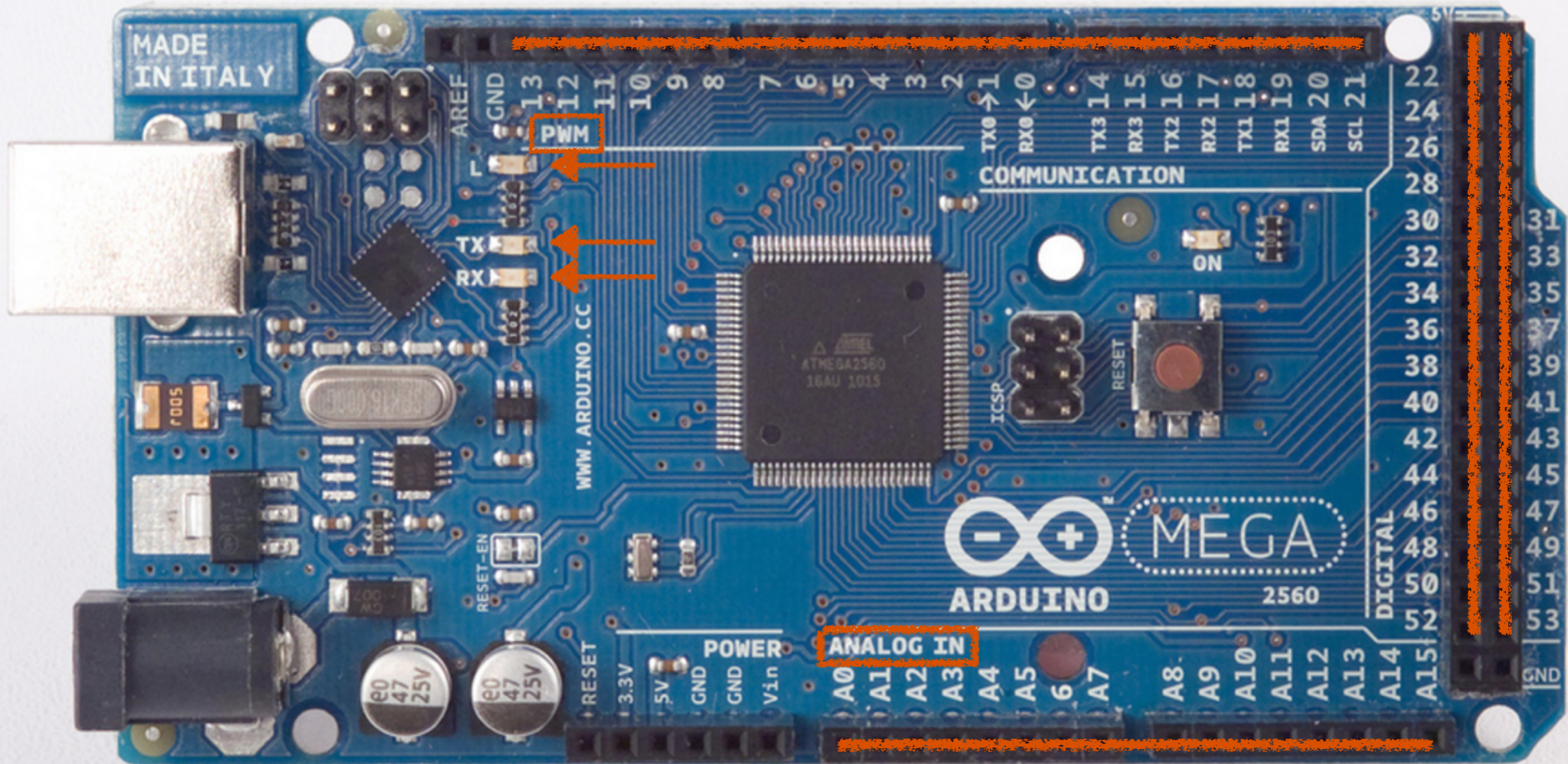
- CPU
- memoria ROM, EPROM, FLASH
- memoria dati: RAM, EEPROM
- oscillatore
- porte I/O
- componenti aggiuntivi: ADC, DAC, contatori, timer...

Arduino Mega2560 (la board che useremo) è composto da un microcontrollore ATmega2560.

DataSheet: <http://www.atmel.com/Images/doc2549.pdf>

Pin I/O

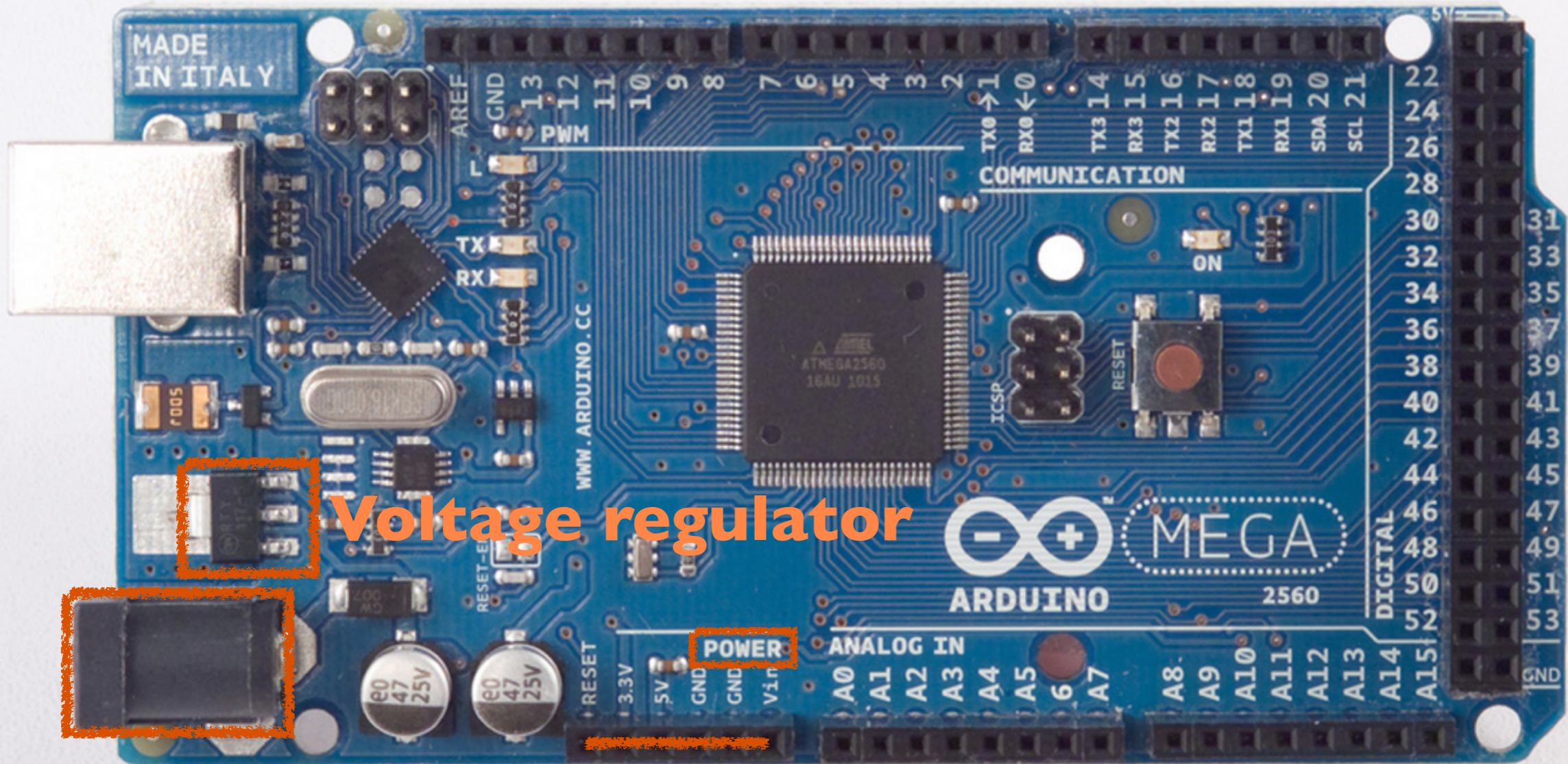
Pin I/O



Pin I/O

- 54 pin in totale, operano a +5 V.
- massima corrente sui pin: 40 mA
- pin 13: LED connesso
- AREF: per input analogici, voltaggio di riferimento.
- Reset
- 16 pin di input analogici: hanno risoluzione di 10 bit (1024 valori)

Power



Voltage regulator

Power Jack

POWER

Voltage Regulator

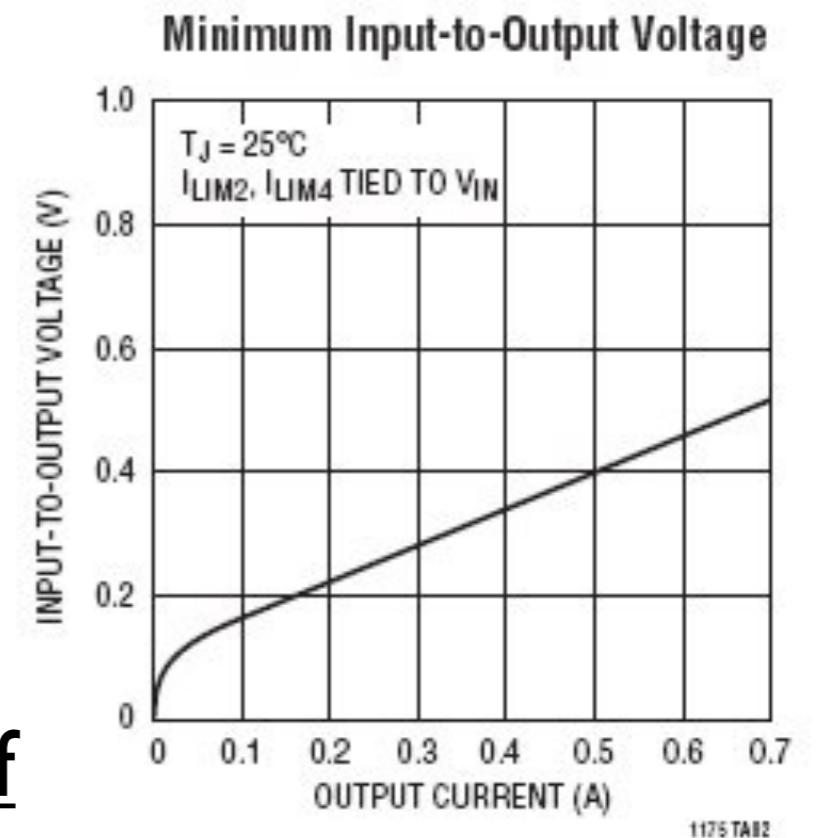
- Serve a stabilizzare la tensione da fornire ai vari componenti della scheda.

- Output: 5V o 3.3V

- Corrente massima fino a 800 mA

- Datasheet:

<http://cds.linear.com/docs/Datasheet/1175ff.pdf>



ICSP e Clock



Oscillatore

ICSP header

Crystal Oscillator

- Circuito oscillante ad alta frequenza (fino ai MHz).
- Basato su oscillazione di un materiale piezoelettrico.
- In questo caso un quarzo, ma può essere anche ceramico.
- Quarzo su Arduino 16 MHz.

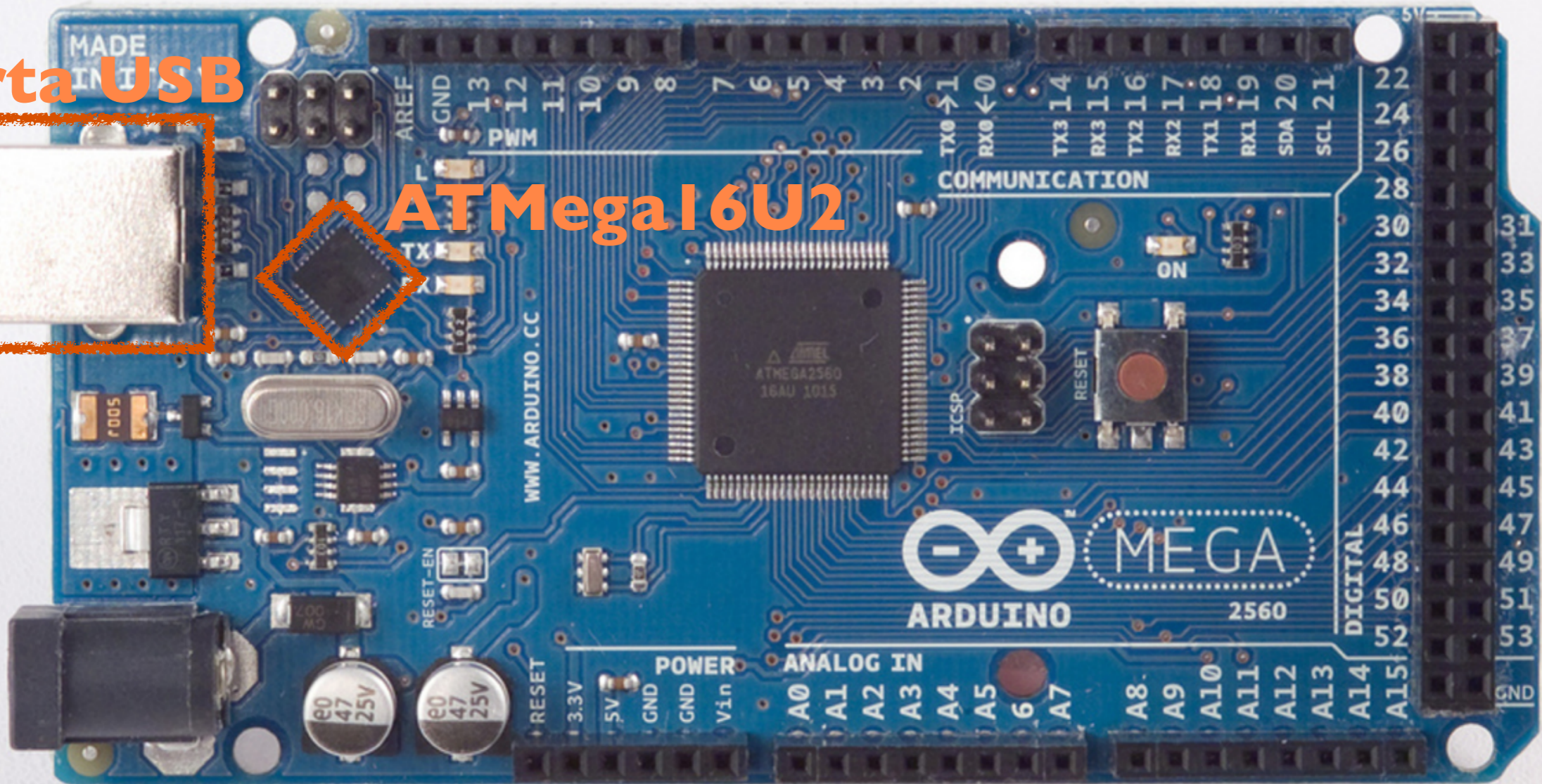
ICSP

- Acronimo per In Circuit Serial Programming
- Può essere usato per programmare Arduino senza passare dal bootloader (non lo useremo).
- Utile se si vuole caricare sulla scheda un programma già sviluppato in un altro ambiente.
- Siccome non usa il bootloader, permette di sfruttare più memoria sulla scheda (8kB su 256kB).

USB port

Porta USB

ATMega16U2



ATMega 16U2

- E' un microcontrollore a tutti gli effetti.
- Meno potente del ATMega2560, è programmato apposta come USB to serial converter.
- Fa le veci del FTDI USB to serial driver, montato sulle vecchie schede.
- DataSheet: <http://www.atmel.com/Images/doc7799.pdf>

Software

Arrays

Arduino

```
int bar[8];  
bar[0] = 1;
```

```
int foo[] = { 0, 1, 2 };
```

Processing

```
int[] bar = new int[8];  
bar[0] = 1;
```

```
int foo[] = { 0, 1, 2 };  
or  
int[] foo = { 0, 1, 2 };
```

Loops

Arduino

```
int i;  
for (i = 0; i < 5; i++) { ... }
```

Processing

```
for (int i = 0; i < 5; i++) { ... }
```

Printing

Arduino

```
Serial.println("hello world");
```

```
int i = 5;  
Serial.println(i);
```

```
int i = 5;  
Serial.print("i = ");  
Serial.print(i);  
Serial.println();
```

Processing

```
println("hello world");
```

```
int i = 5;  
println(i);
```

```
int i = 5;  
println("i = " + i);
```

- Arduino è programmabile.
Ambiente di sviluppo dedicato.

- Linguaggio simile al C/C++.

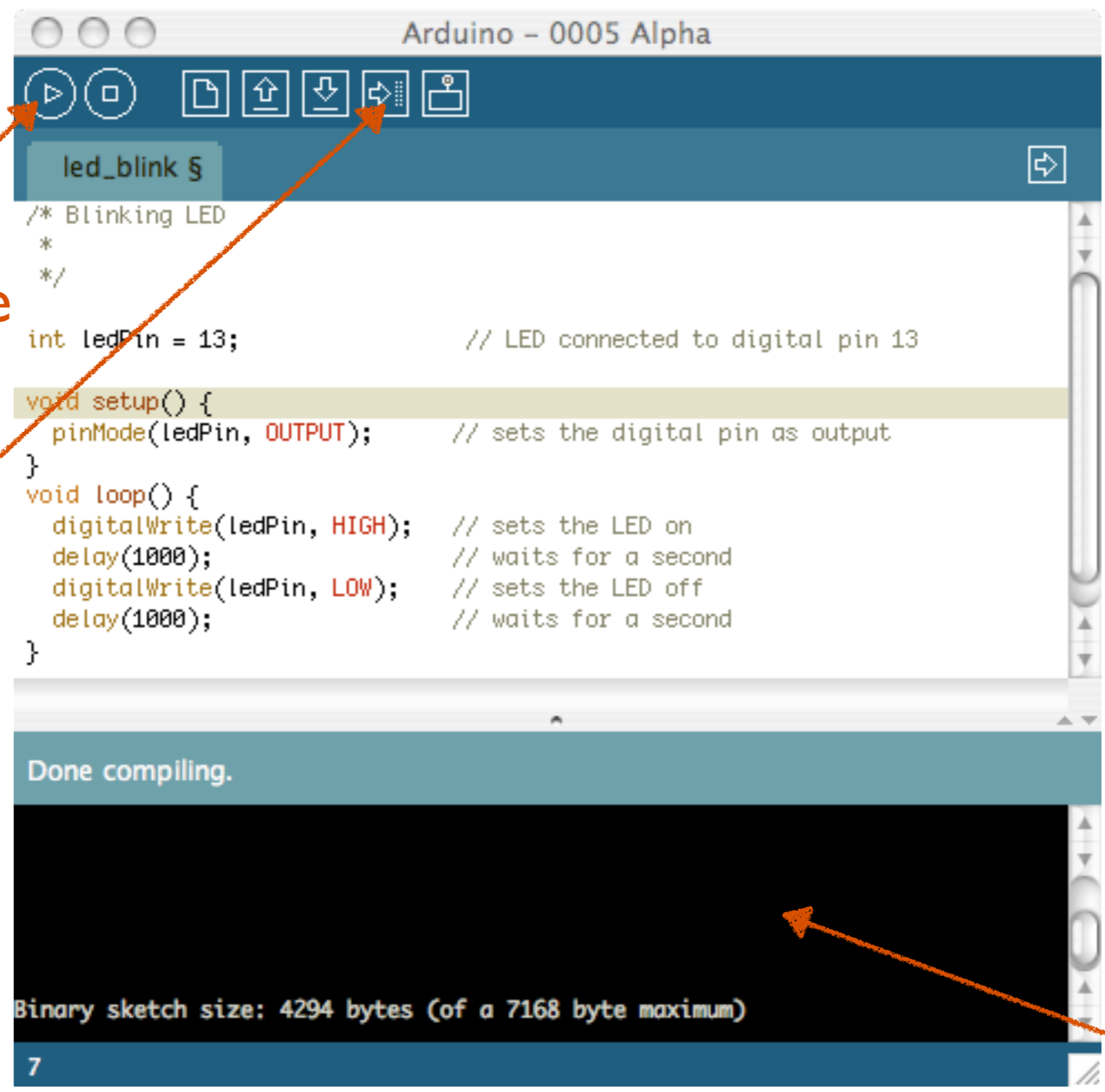
- Libreria Wiring per gestire input/output

- Derivato da Processing ma con alcune differenze.

Software

compilazione

upload



status

Software

Funzioni principali: **setup()** e **loop()**

setup(): chiamata una sola volta, all'inizio del programma. Usata per inizializzazione delle variabili e dello stato dei pin.

loop(): è un ciclo, all'interno del quale implementare il vero e proprio programma

Librerie e funzioni predefinite per accedere ai pin e impostare/leggere il valore.

```
int buttonPin = 3;

// setup initializes serial and the button pin
void setup()
{
  beginSerial(9600);
  pinMode(buttonPin, INPUT);
}

// loop checks the button pin each time,
// and will send serial if it is pressed
void loop()
{
  if (digitalRead(buttonPin) == HIGH)
    serialWrite('H');
  else
    serialWrite('L');

  delay(1000);
}
```

Installazione

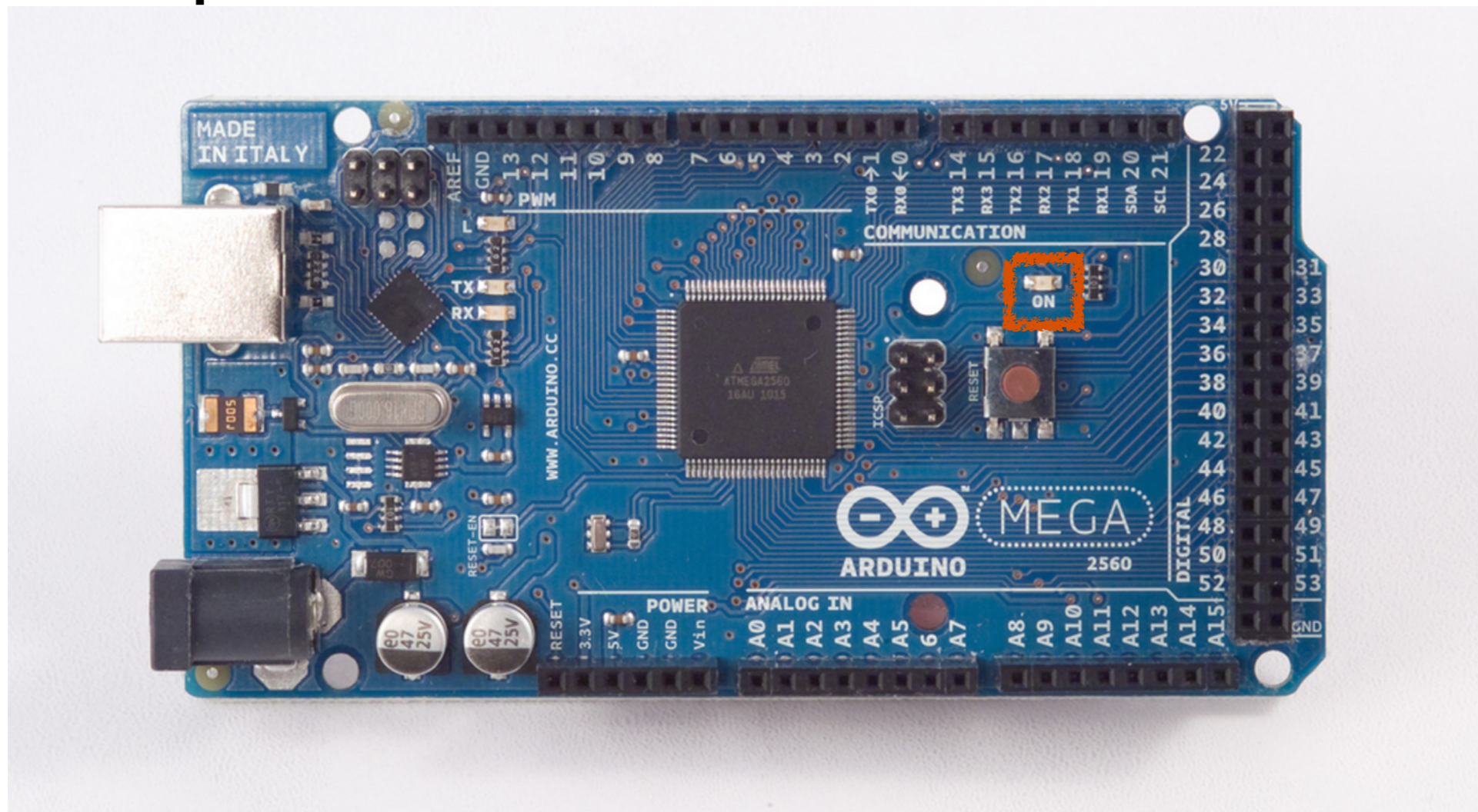
Passo 1: scaricare il software da qui

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

Installazione

Passo 2: connettere la board al pc tramite USB.

A questo punto il LED **PWR** dovrebbe accendersi.



Installazione

Passo 3: installare i driver per Arduino.

- 1) cliccare su Start Menu
- 2) Control Panel
- 3) System and Security
- 4) System
- 5) Device Manager
- 6) Ports (COM & LPT)
- 7) cliccare col destro su Arduino e scegliere Update Driver Software
- 8) scegliere “browse my computer” e selezionare il file .inf appena scaricato (sotto la cartella Drivers).

Installazione

Passo 4: lanciare l'applicazione.

Passo 5: aprire l'esempio blink.

Andare sotto File, Examples, 1.Basics, Blink.

Passo 6: selezionare il tipo di Board.

Andare sotto Tools, Board e selezionare
Arduino Mega 2560.

Installazione

Passo 7: selezionare la porta seriale.

Andare sotto Tools, Serial Port. Cliccare su COM3. Per trovare quale è basta disconnettere e riconnettere la Board.

Passo 8: caricare il programma.

Basta cliccare sul bottone Upload. I LED RX e TX dovrebbero iniziare a lampeggiare. Se l'upload riesce compare il messaggio "Done Uploading".

Dopo pochi secondi dall'upload dovrete vedere il LED I3 che si accende ripetutamente.

Disclaimer

10 metodi certi per distruggere Arduino:

- 1) Cortocircuitare I/O pin a terra.
- 2) Connettere I/O pin tra di loro.
- 3) Applicare troppa tensione sugli I/O pin.
- 4) Applicare tensione su V_{in} , ma a polarità inversa.
- 5) Applicare $>5V$ sul pin 5V.
- 6) Applicare $>3.3V$ sul pin 3.3V.
- 7) Cortocircuitare V_{in} a GND.
- 8) Applicare $>13V$ al reset.
- 9) Applicare tensione a 5V e caricare V_{in} .
- 10) Eccedere la corrente totale del microcontrollore (200 mA).

Per saperne di più: <http://ruggedcircuits.com/html/ancp01.html>

Sketch 1: blinking LED

```
/*  
  Blink
```

Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.

```
*/
```

```
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino  
boards:  
  pinMode(13, OUTPUT);  
}
```

Setup()

```
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);           // wait for a second  
}
```

Loop()

Sketch I: blinking LED

Proviamo a modificare il programma di test.

- 1) modificare la frequenza del LED
- 2) aggiungere altri LED (modifica Pin I/O)
- 3) fare accendere diversi LED in sequenza
- 4) Dare un'occhiata allo sketch Fading LED
- 5) aumentare o ridurre la brightness del LED.
- 6) ...

Bibliografia

www.arduino.cc

<http://arduino.cc/en/Tutorial/HomePage>

<http://arduino.cc/en/Reference/HomePage>

<http://arduino.cc/en/Guide/Environment>

<http://todbot.com/blog/spookyarduino/>

<http://todbot.com/blog/bionicarduino/>

<http://wiring.org.co/>

<http://processing.org/>