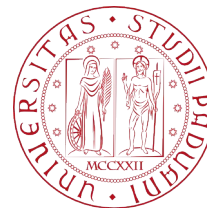


Introduction to Arduino

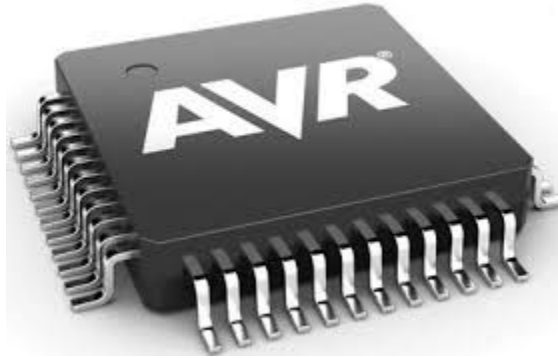
Mobile Computing, aa. 2016/2017
May 12, 2017

Daniele Ronzani - *Ph.D student in Computer Science*
dronzani@math.unipd.it



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

What are Microcontrollers

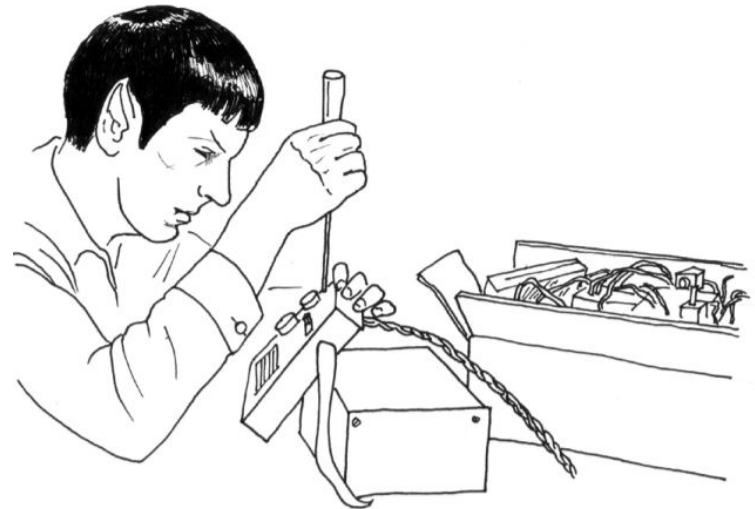


- Very small and simple computers
- Cheap and useful
- Present in every smart system
- **Atmel** is one of the family of microcontrollers

What is the DIY paradigm



- Create something without professional knowledge
- Many fields
 - Programming
 - Education
 - Critical theory
 - Business
 - **Electronics**
 - ...



The Arduino project



Arduino UNO

***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.*

Arduino.cc

- An Italian prototyping tool project
- Ready-made electronic products at low costs
- Open Source
- Boards with different performances
- Shields that extend the functions
 - Wi-Fi, GPSR, Ethernet, microSD reader, ...



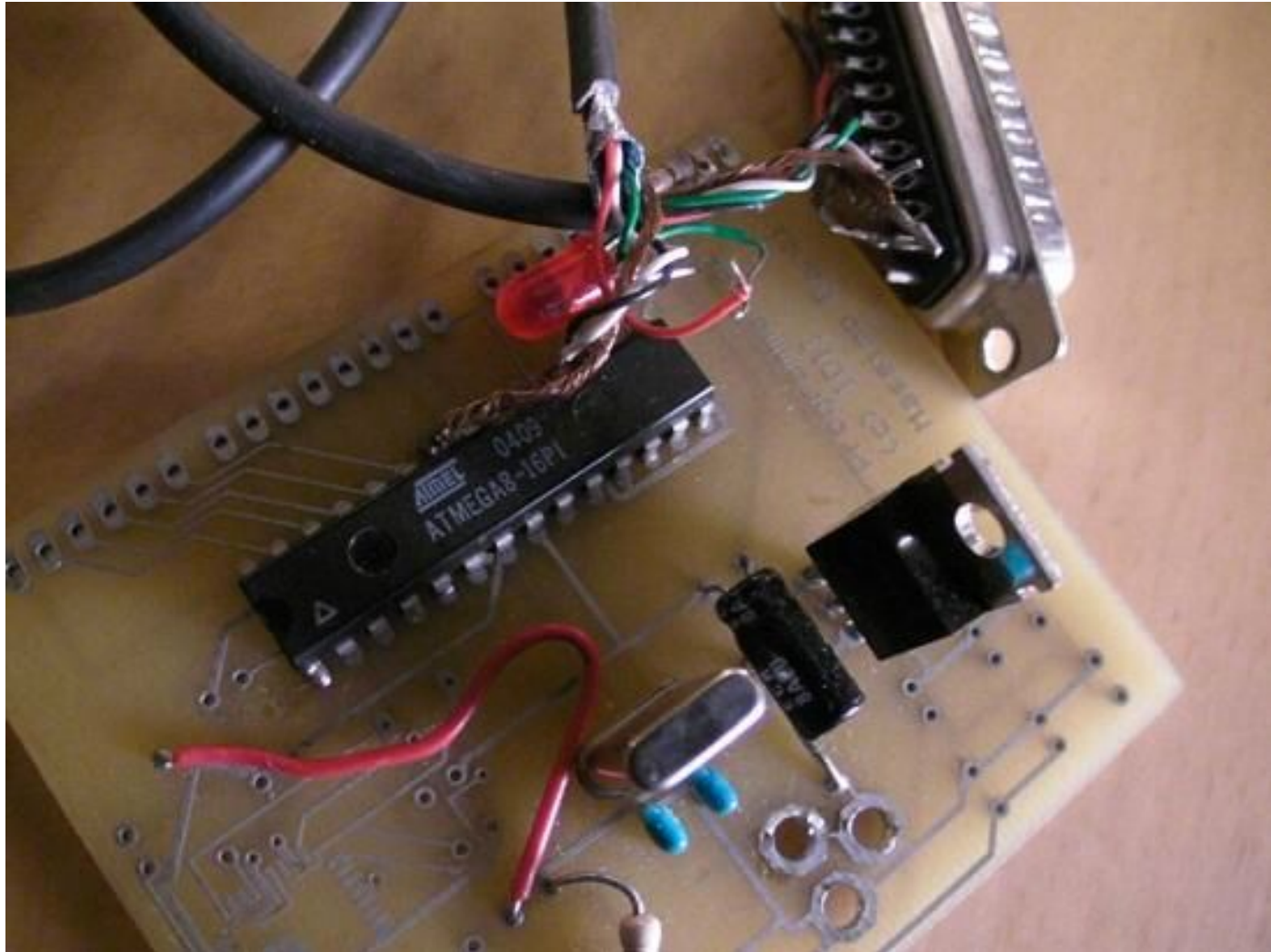
Shields and sensors

- Arduino was born in 2005 in Ivrea, Italy
- **IDEA:** *less expensive device for controlling interactive electronic projects*
 - *Rapid prototyping*
- Name "*Arduino*" in honor of *Bar di Re Arduino*
- Start to produce boards in a small factory
- Arduino was built around the **Wiring** project of Hernando Barragan.
 - Hernando's thesis project at the Interaction Design Institute Ivrea.
 - Remove the deep knowledge barriers



Massimo Banzi

The First Prototype



Arduino Today



Google Trends - Web Search Interest: arduino - Worldwide, 2004 - present

Search terms ?

Arduino

+ Add term

Other comparisons

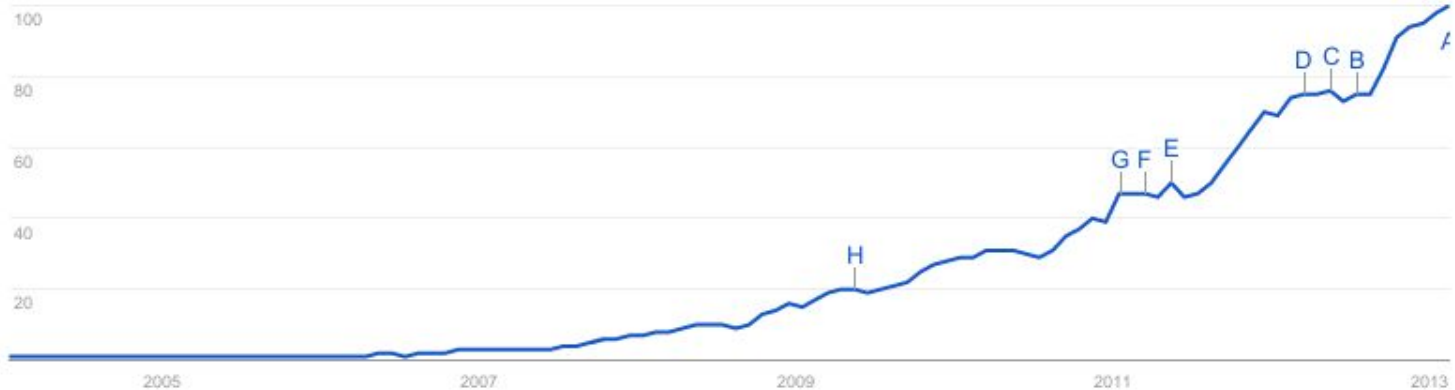
Limit to

Web Search

Worldwide

2004 - present

All Categories



Arduino community is mushrooming across the globe

Embed

Regional interest ?



80% from
Europe and USA

More than 200
distributors



Related terms ?

Top

Rising

arduino uno	100	<div></div>
arduino shield	80	<div></div>
arduino projects	75	<div></div>
arduino serial	70	<div></div>
arduino led	70	<div></div>
arduino tutorial	50	<div></div>
lcd arduino	50	<div></div>



● Lego Mindstorm

- Intelligent brick computer
- Programmed in Brick Logo
- Strong community
- Too much expensive (cost: ~ \$349)

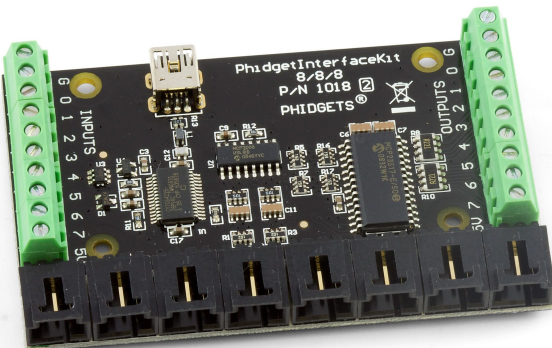


● i-cubeX

- Objects respond to human actions and environmental parameters
- Human Interface Devices
- Easy but expensive

● Phidgets

- Easy-to-assemble sensors and controllers
- Not open source (proprietary)
- Requires less hardware and software knowledge
- ~ 3.5 x the cost of Arduino board



- **Make Controller Kit**

- More powerful CPU than Arduino
- Has quite a lot of "*bang for the buck*"

- **Raspberry**

- It is closer to a computer
- Includes CPU, USB ports, Ethernet, HDMI, ...
- Includes an own OS (Linux)

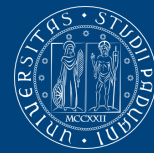
- **Teensy**

- Same Arduino firmware
- Compatible with Arduino
- Very small size

- **UDOO**

- Merges Arduino and Raspberry
- Powerful prototyping platform
- Linux or Android OSs

The three key concepts of Arduino Project



Hardware

Can sense the environment by sensors, and affects it by controlling lights, motors, and other actuators.



Environment

Allows to write code in the Arduino programming language and using the Arduino development environment.

```
Blink | Arduino 1.0.3

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
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 (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

Community

It is made up of everyone from and to hobbyists, students, designers and engineers all across the world.

Using Arduino		
Installation & Troubleshooting For problems with Arduino itself. NOT your project Last post: Today at 10:51 am Re: Installable input using... by danyal808	76,885 Posts	18,037 Topics
Project Guidance Advice on general approaches or feasibility Last post: Today at 10:53 am Re: Arduino Bluefruit LE UART... by Nemo_Franks	378,669 Posts	51,792 Topics
Programming Questions Understanding the language, error messages, etc. Last post: Today at 10:53 am Re: Beginner RGB LED pins... by subcanny	519,563 Posts	63,385 Topics
General Electronics Resistors, capacitors, breadboards, soldering, etc. Last post: Today at 10:54 am Re: 555 and ceramic caps by Mark7	180,280 Posts	17,845 Topics
Microcontrollers Standalone or alternative microcontrollers, in-system programming, bootloaders, etc. Last post: Today at 09:27 am Re: Using arduino UNO as... by tetrog001	74,009 Posts	8,552 Topics
LEDs and Multiplexing Controlling lots of inputs and outputs Last post: Today at 09:26 am Re: How to make certain... by negatig	52,185 Posts	6,492 Topics
Displays		

Why should we use Arduino?

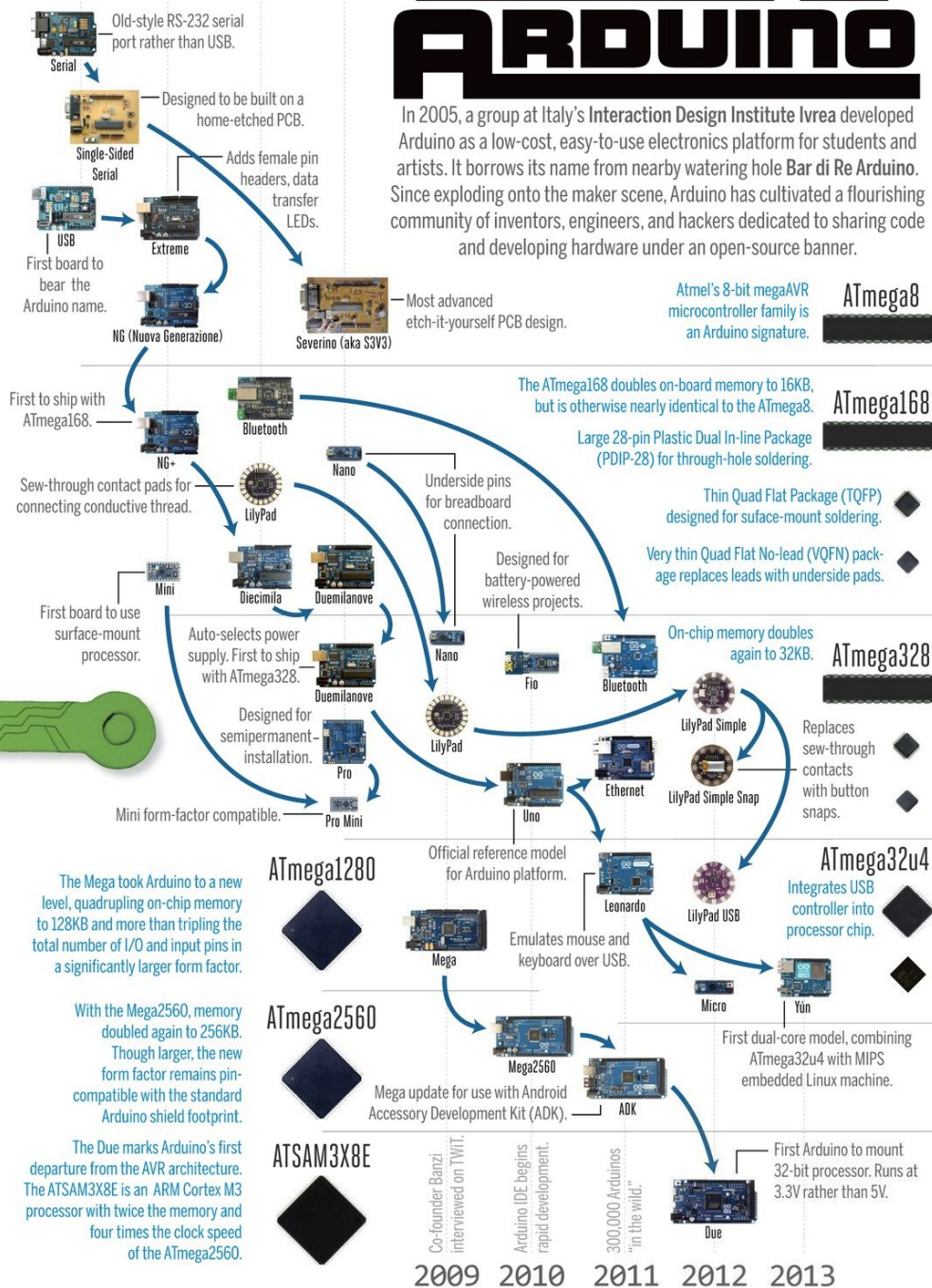


- **Simple, academic purposes**
- **Open source**
 - **Hardware & Software:** permits to manufacture the boards and software distribution by anyone
 - Arduino compatibile: Canaduino, Freeduino, Linduino, SainSmart, ...
 - GNU Lesser General Public License (LGPL)
- Opportunistic **prototyping**
- **Community**
 - Wiki
 - Forum
 - Tutorials
- Could be used as an **IoT starting point**
 - Physical computing objects

2005 2006 2007 2008

EVOLUTION OF ARDUINO

In 2005, a group at Italy's Interaction Design Institute Ivrea developed Arduino as a low-cost, easy-to-use electronics platform for students and artists. It borrows its name from nearby watering hole Bar di Re Arduino. Since exploding onto the maker scene, Arduino has cultivated a flourishing community of inventors, engineers, and hackers dedicated to sharing code and developing hardware under an open-source banner.



- 2006: New Generation (NG)
 - ATmega168
 - First Mini Arduino
- 2007: Portable and communication
 - i-Bluetooth
 - Sew-through contacts pads
- 2009: Nano and LilyPad
 - New Arduino Mega with ATmega1280
- 2010: Rapid development
 - **Official Arduino Uno**
- 2012: Miniaturization and CPU power enhance
 - LilyPad Simple/Snap and Micro
 - Arduino Due (32 bit)

Evolution of Arduino



- 2013: 700.000 official boards were sold
- 2016: **17** versions of the Arduino board have been commercially produced.



Arduino Uno



Arduino Leonardo



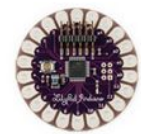
Arduino Mega ADK



Arduino Ethernet



LilyPad Arduino
SimpleSnap



LilyPad Arduino



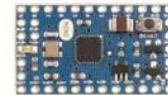
Arduino Due



Arduino Yún



Arduino Mega 2560



Arduino Mini



Arduino Nano



Arduino Pro Mini



Arduino Tre



Arduino Micro



LilyPad Arduino USB



LilyPad Arduino
Simple

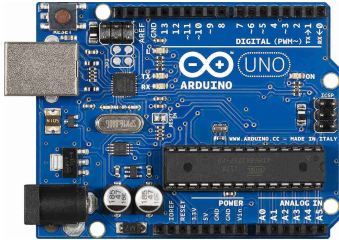


Arduino Pro



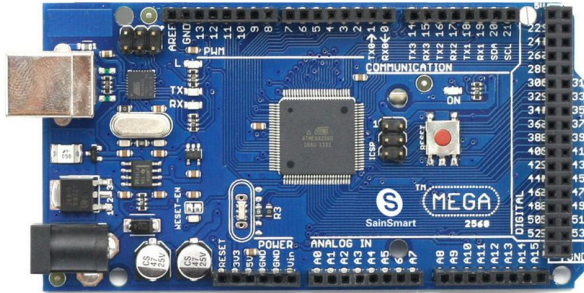
Arduino Fio

Some Current Arduino Boards



UNO

- Current official reference of Arduino Boards
- Most used and documented board



Mega

- Designed for more complex projects
- 54 digital I/O pins, 16 analog inputs
- ATmega2560



LilyPad

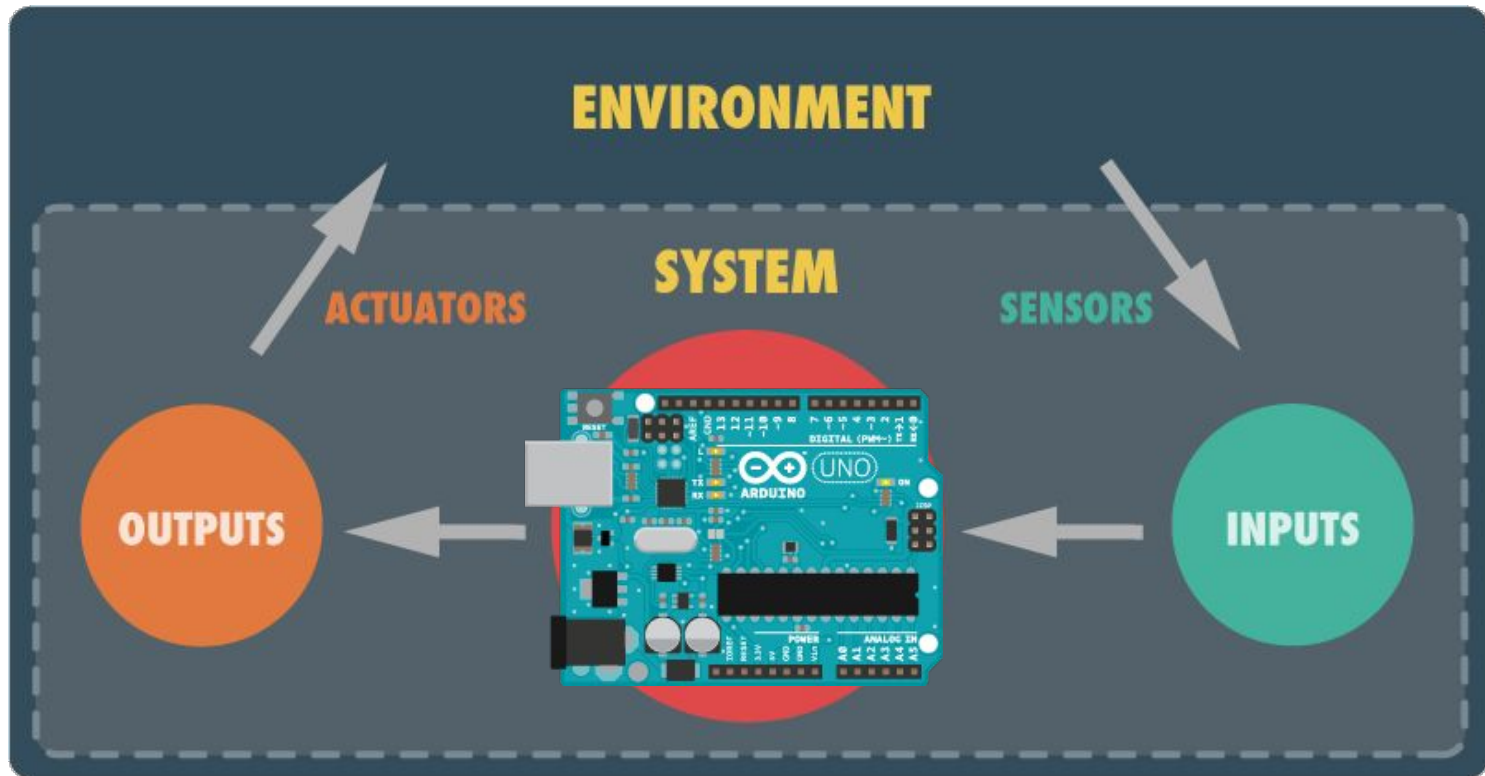
- Designed for e-textiles and wearables projects
- Can be sewn to fabric and to power supplies



Nano

- Compact board similar to the UNO

Arduino as Physical Computing Platform



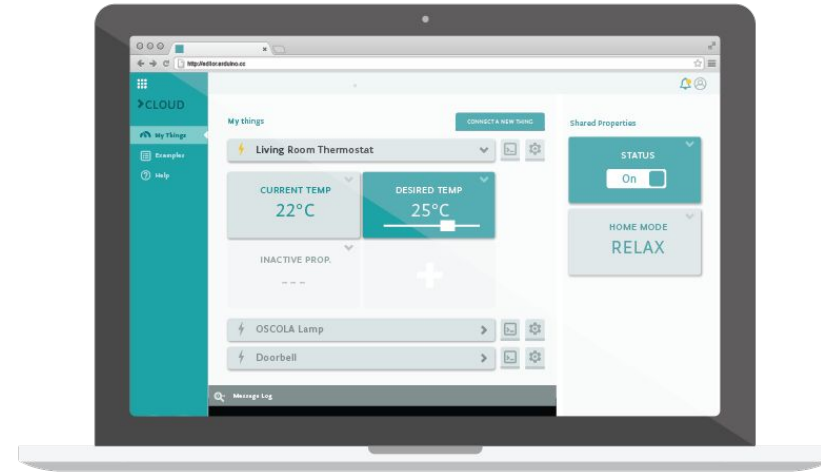
Physical Computing involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behaviour implemented as software running inside a microcontroller.



IoT is a new emerging technology where the things are connected altogether and interact

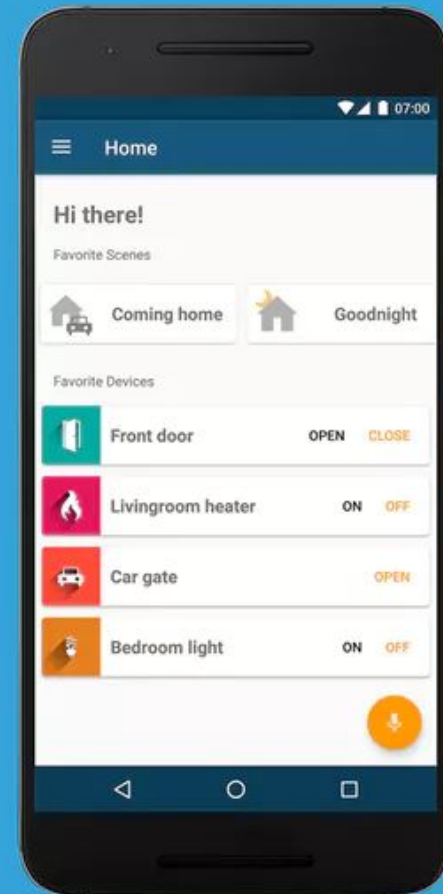
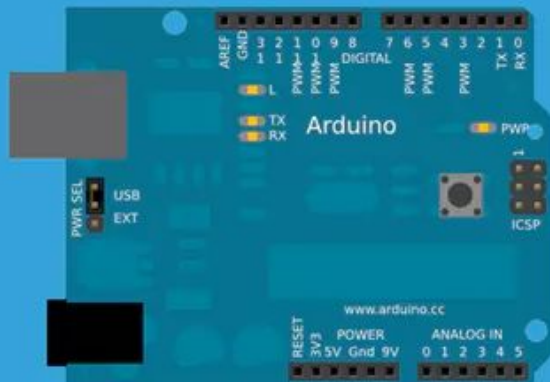
- Arduino provides the tools
 - Remote processing of data
 - Connection of physical objects to the Internet
- IoT Manifest
 - **Open SW & HW:**
 - Possibility to share works
 - More innovation
 - **Sustainable**
 - Devices should be easily upgraded
 - **Fair**
 - You should have control of your devices
 - Security



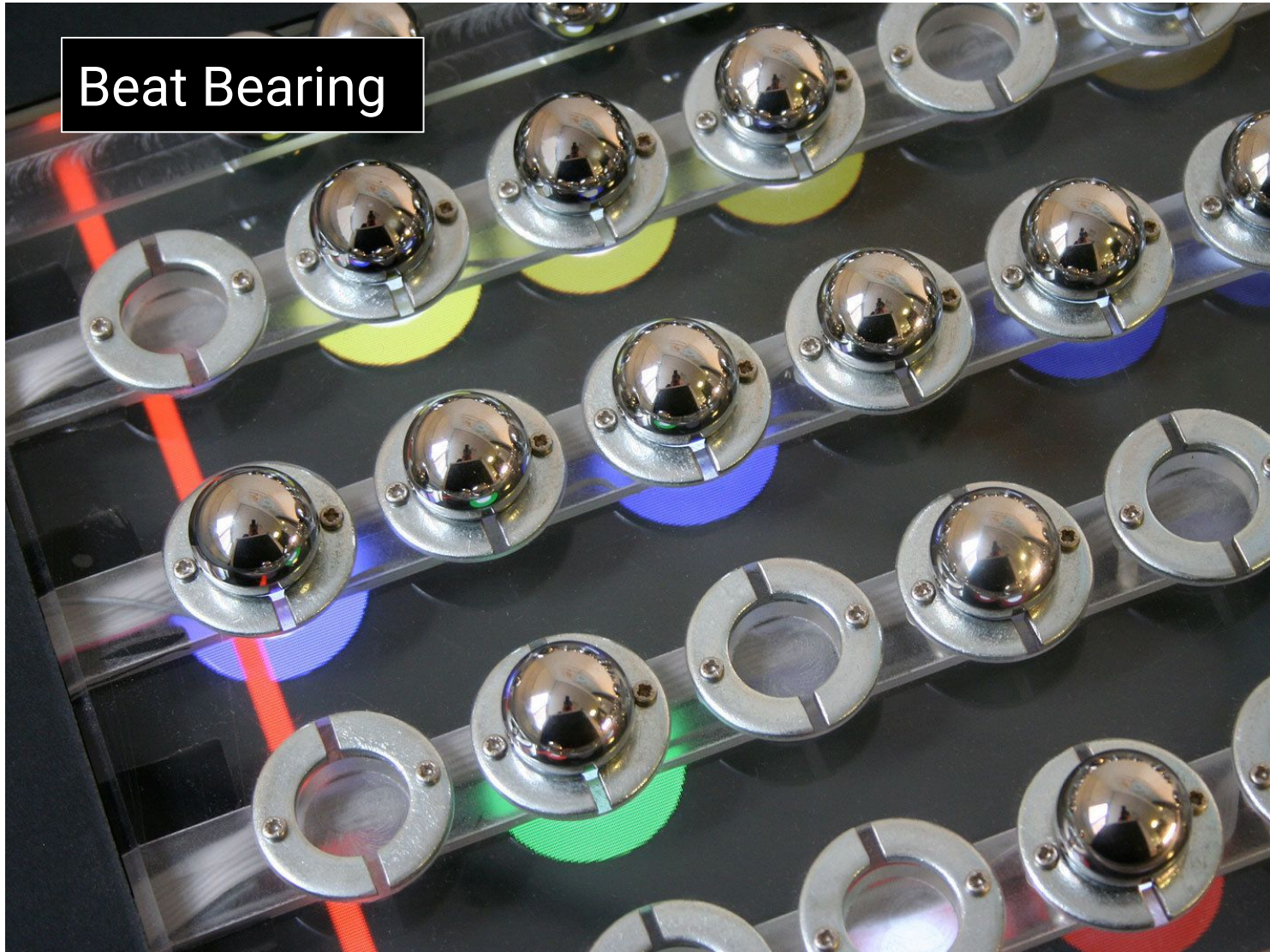


- A new platform to make building IoT
- Creation of tools that allow connection and control of device on/by the Internet
- MQTT broker makes connection between each object

Arduino & Android Home remote control



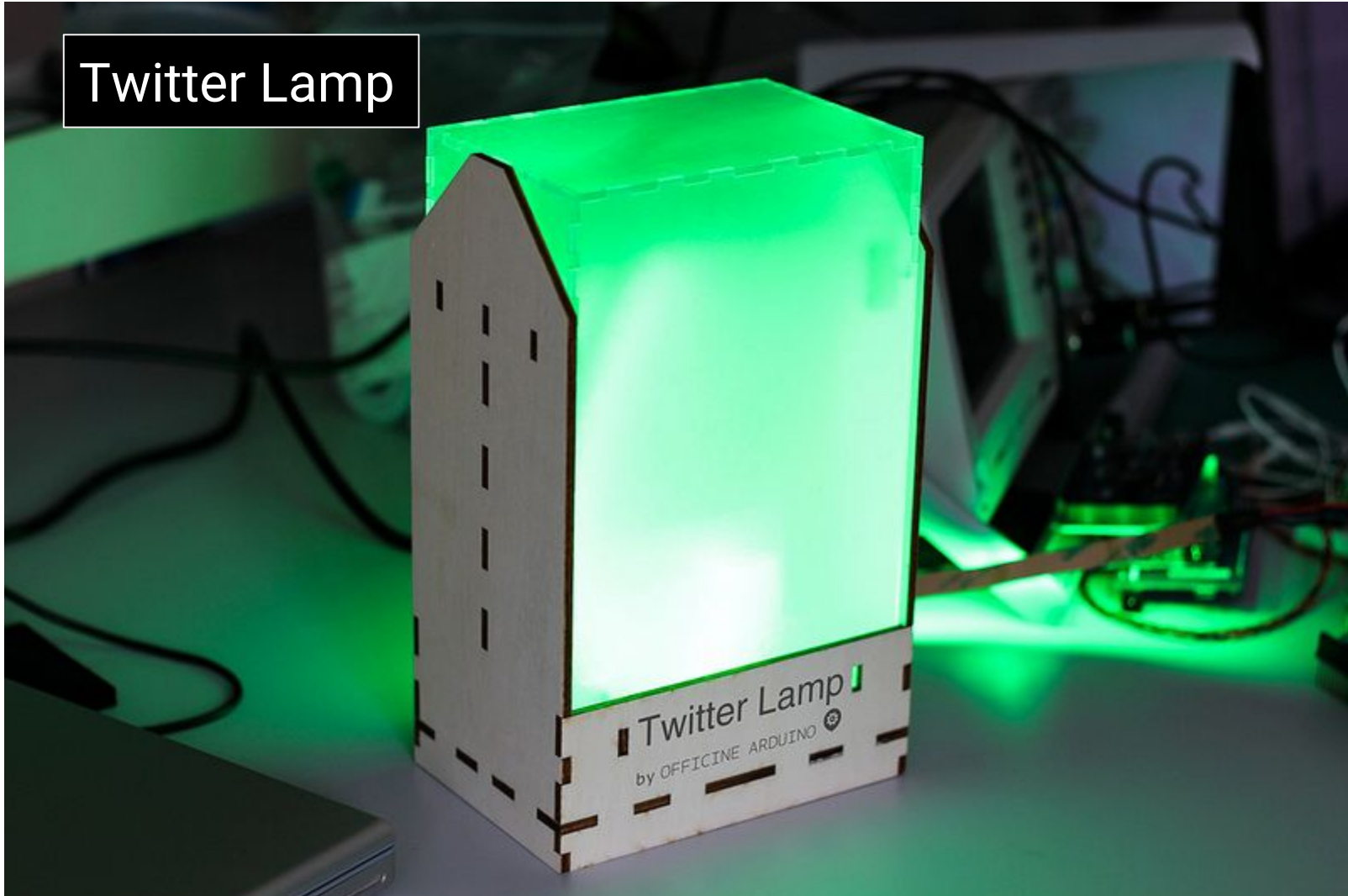
Beat Bearing



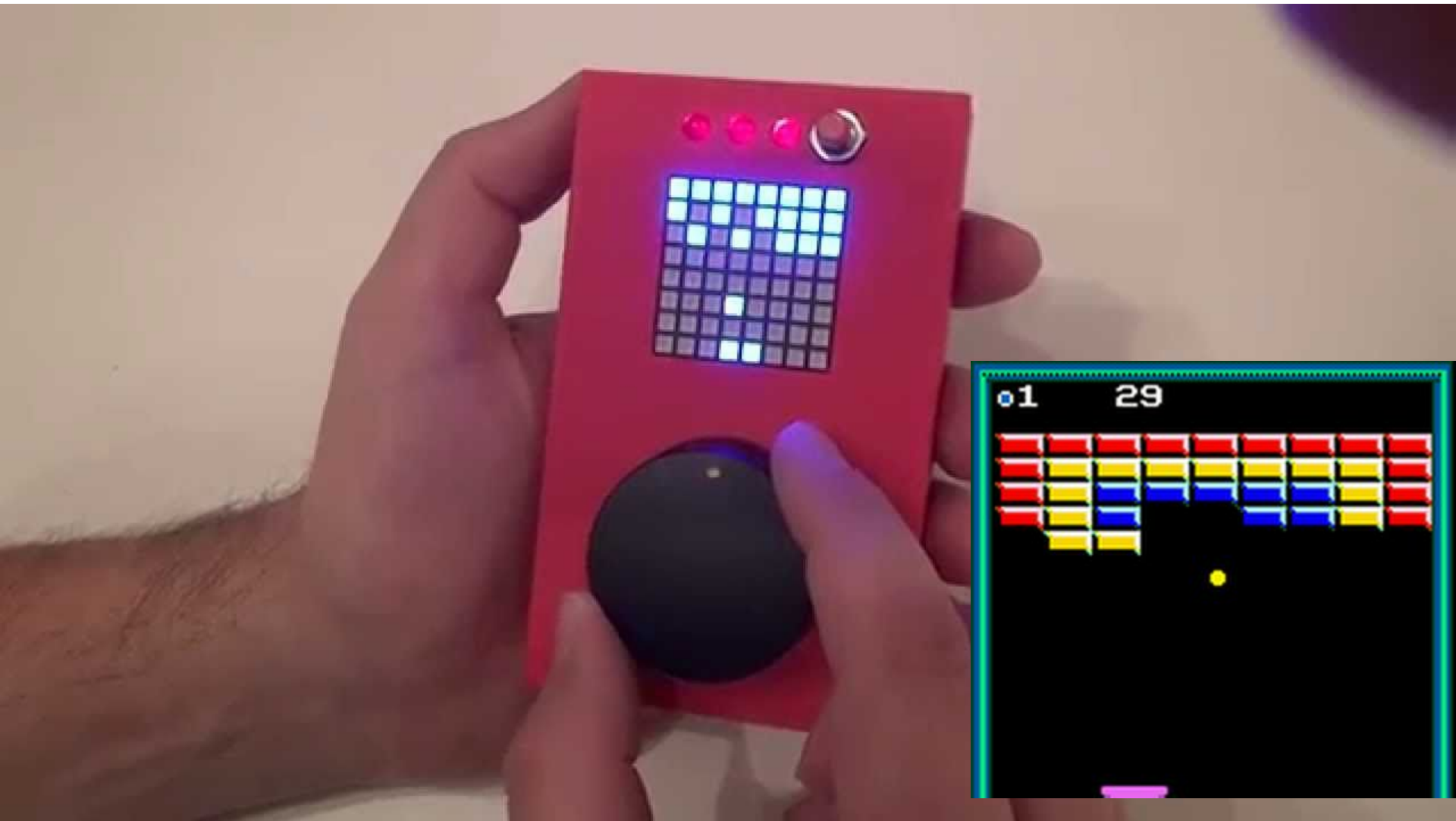
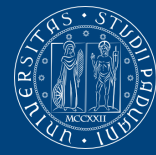
Arduino Usage Examples



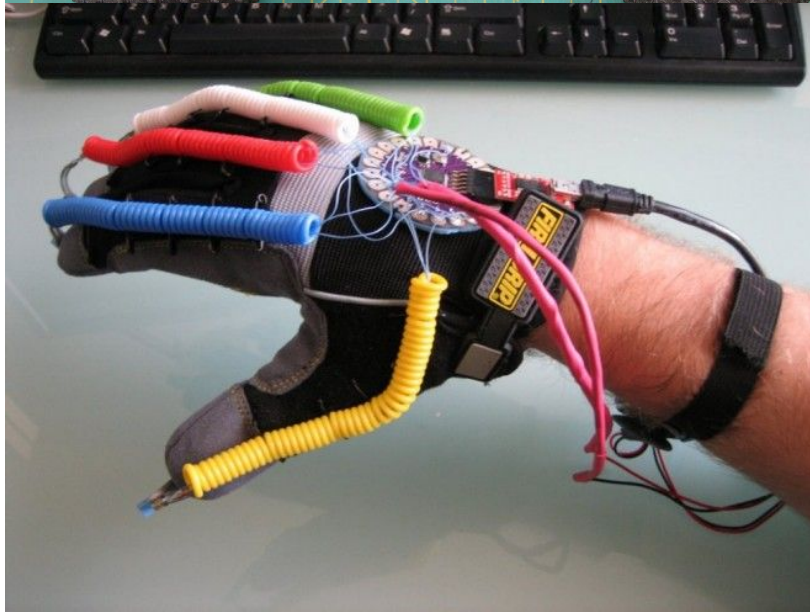
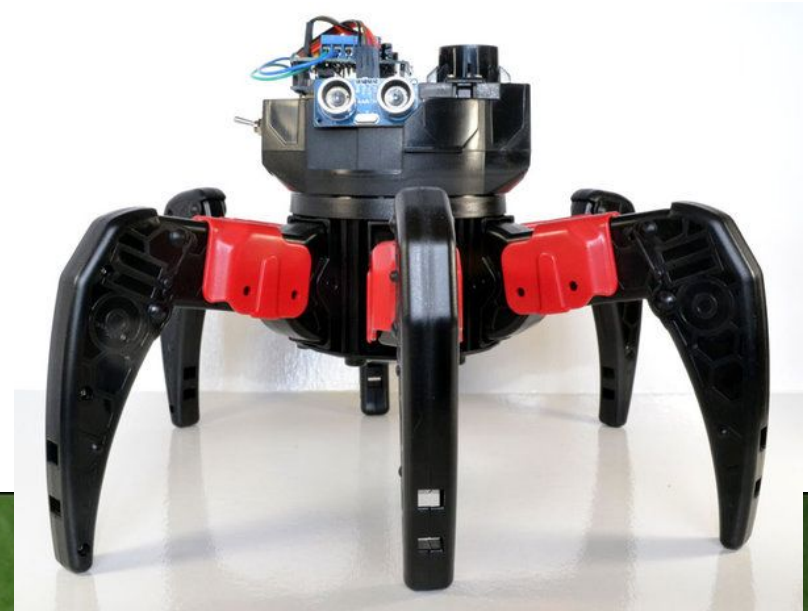
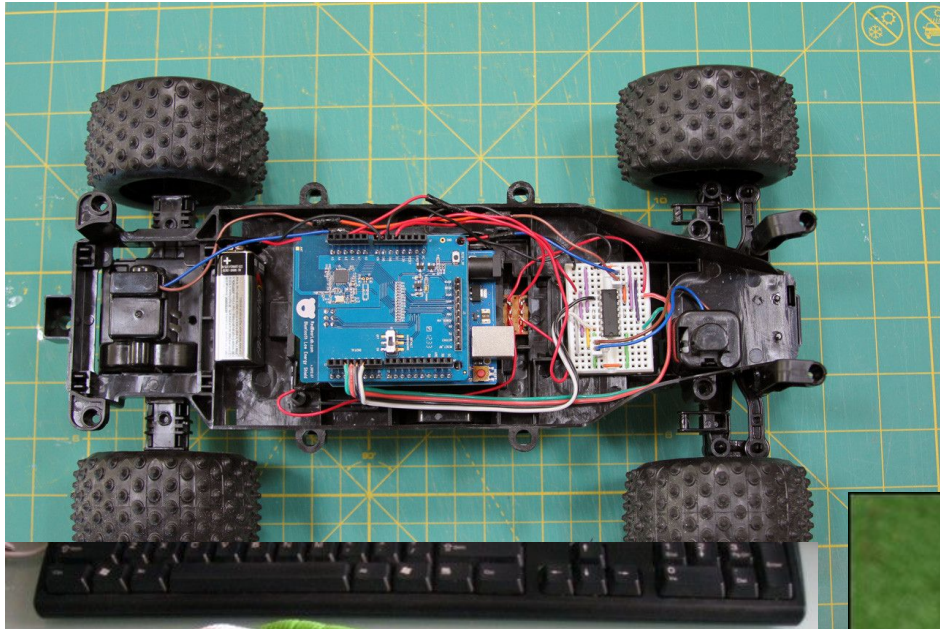
Twitter Lamp



TeleBall (Breakout game)



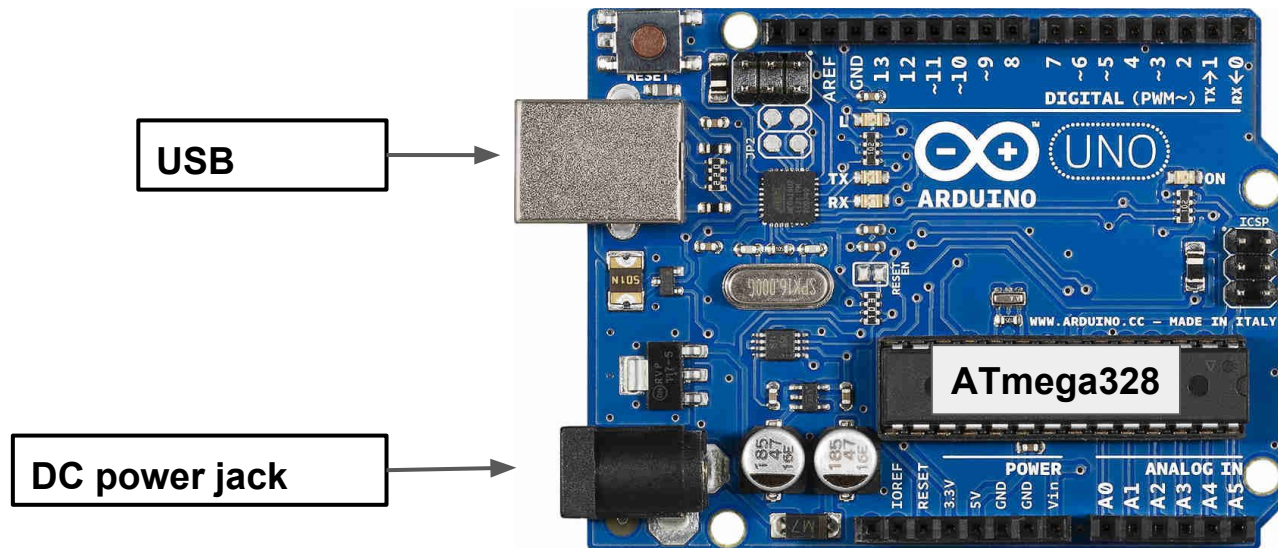
Other Examples



Arduino Architecture and Components

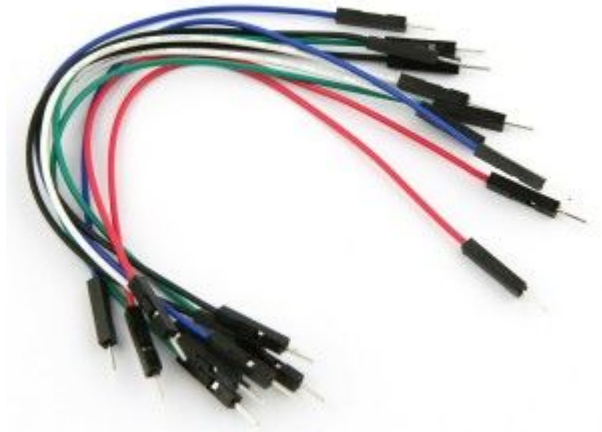
Main components

- AVR Microcontroller
- Analog and digital I/O pins
- Flash memory
 - Integrated in the microcontroller
- USB port for serial communication



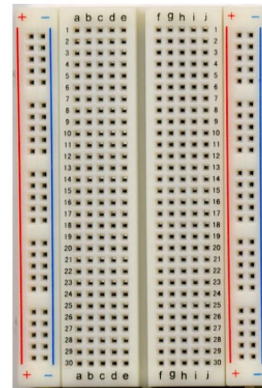
Components: mandatory

Wires



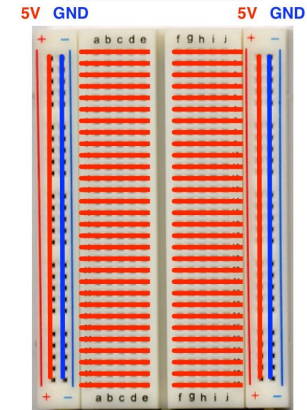
Prototyping board (breadboard)

Breadboard (photo)



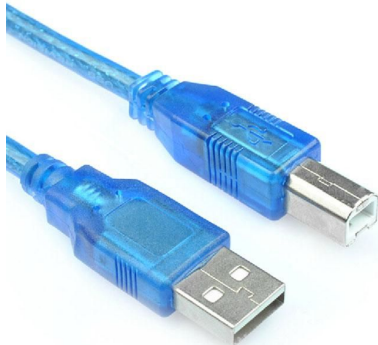
Video and Website © 2004 ClarkZapper.net

Breadboard (schematic)

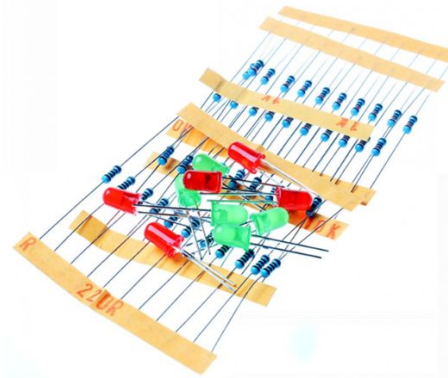


Video and Website © 2004 ClarkZapper.net

USB connector



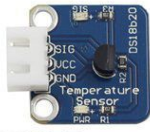
Resistors and leds



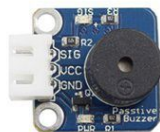
Components: I/O devices



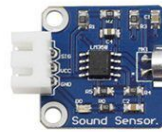
Barometer



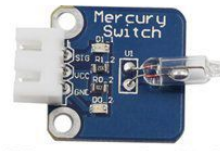
DS18B20 Temperature



Passive buzzer



Sound Sensor



Mercury Switch



Rotary Encoder



Tracking sensor



Auto-flash LED



Dual-color LED

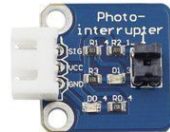


Photo-interrupter



Switch Hall



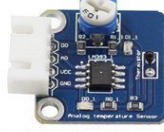
Obstacle Avoidance



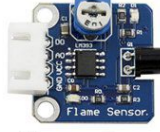
RTC-DS1302



Ultrasonic



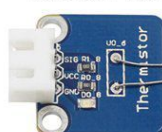
Analog temperature



Flame Sensor



Photoresistor



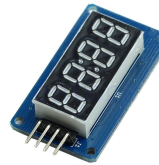
Thermistor module



ADDA Converter



Remote Controller



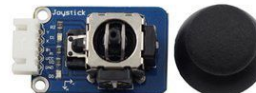
Analog Hall



Humiture sensor



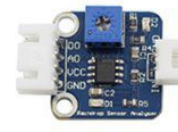
Potentiometer



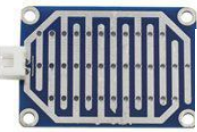
Joystick PS2 module



Color Sensor



Raindrop Sensor



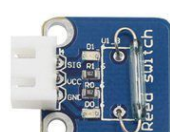
Tilt Switch



Active buzzer



Infrared-Receiver



Reed Switch



MQ-2 Gas Sensor



Relay Module



Button module



Laser Transmitter



RGB LED



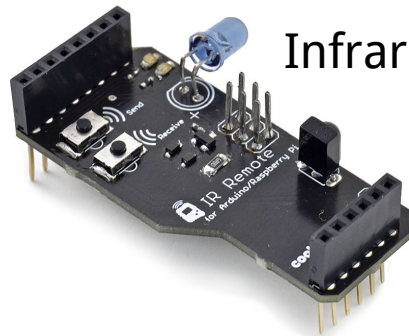
Touch Switch



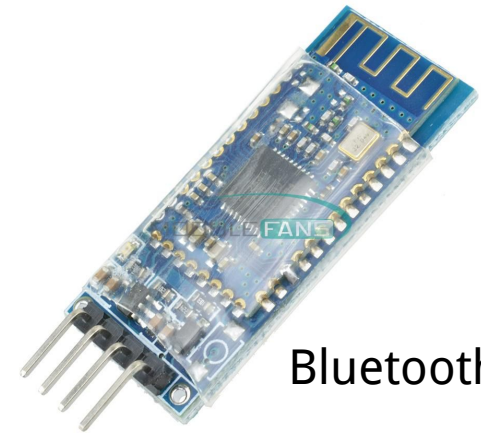
Wired and Wireless comm. devices



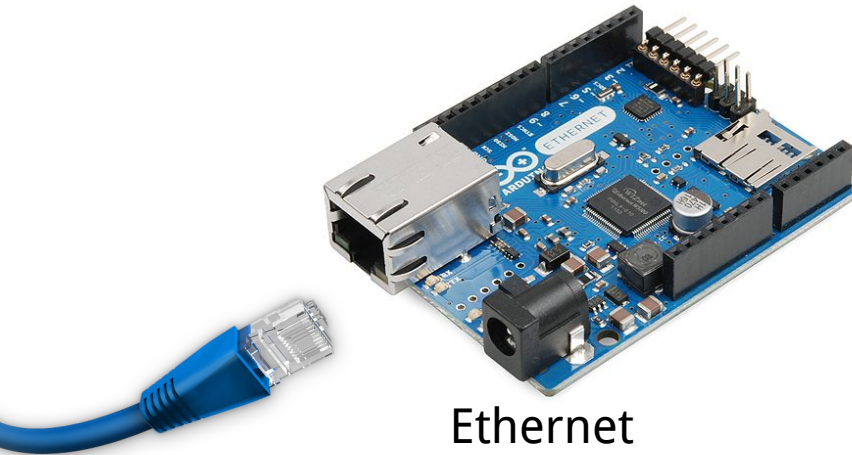
WiFi



Infrared



Bluetooth



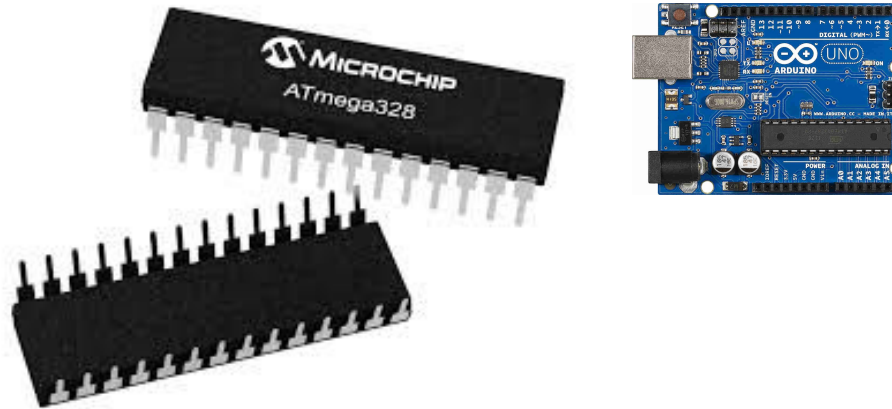
Ethernet



Zig-Bee



Arduino UNO (ATmega 328)

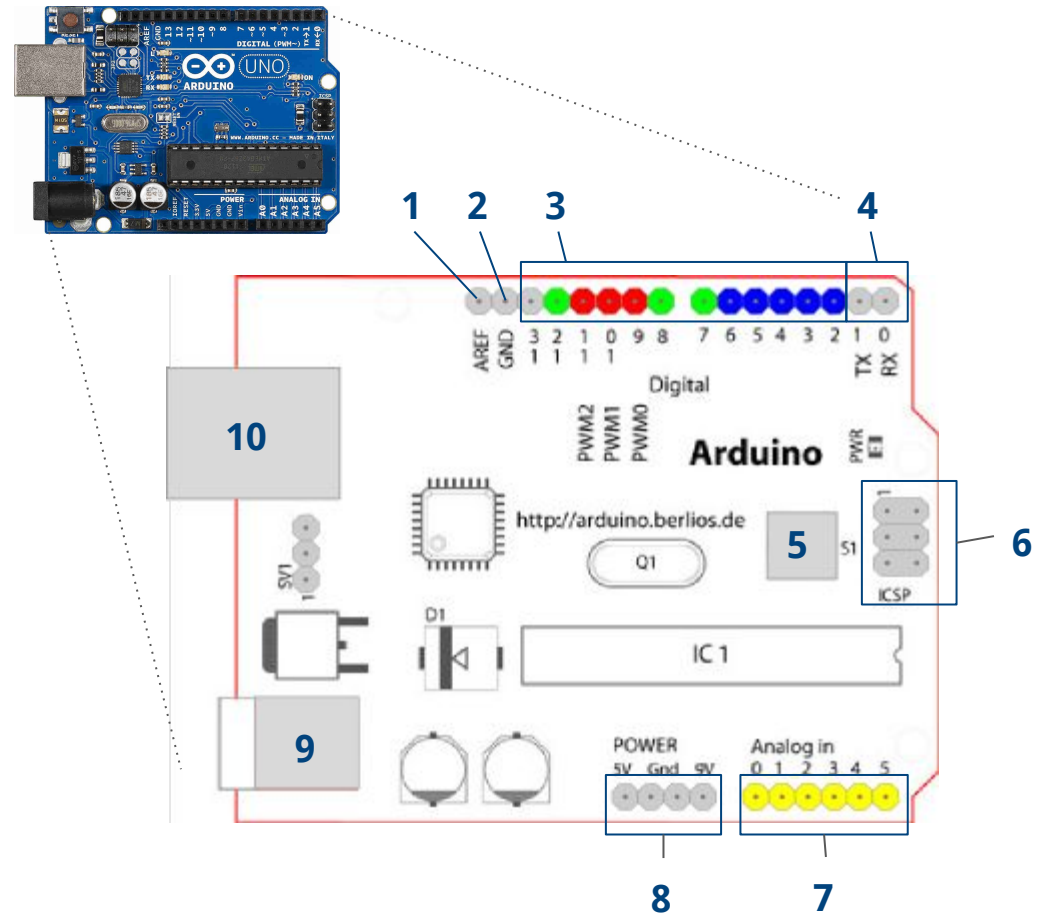


Technical characteristics

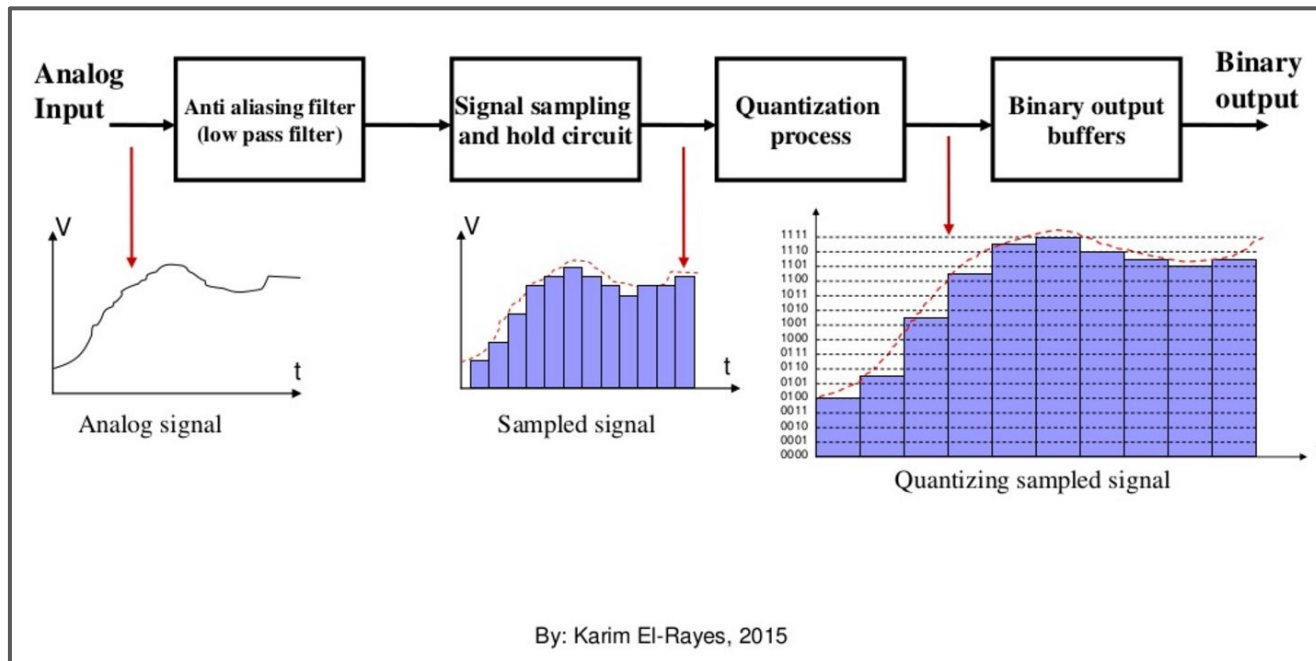
- Clock speed: 16 MHz (*Intel 286*: 12.5 MHz) - 8-bit
- Flash program memory: 32 KBytes (0.5 used by bootloader)
- SRAM: 2 KBytes
- Input / Output
 - 14 digital input/output pins
 - 6 analog input pins
 - 6 analog output pins (PWM)

Layout of Arduino UNO

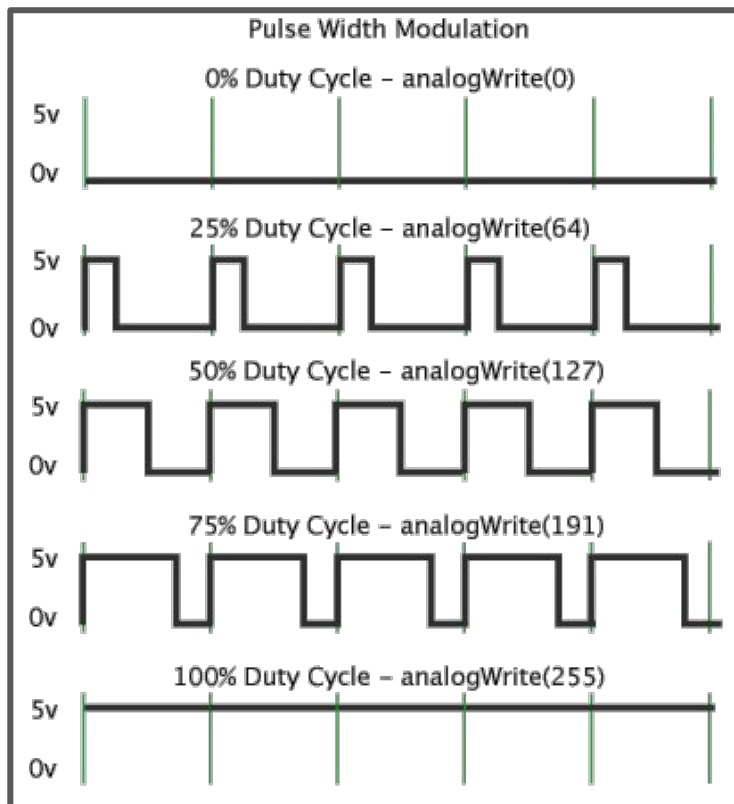
1. Analog Reference pin
2. Digital Ground
3. Digital pins 2-13
4. Digital pins 0-1/Seral In/Out
5. Reset Button
6. In-circuit Serial Programmer
7. Analog In pins 0-5
8. Power and Ground pins
9. External Power Supply In
10. USB port



- Digital pins have only two values on/off (0/1)
 - Arduino UNO has 14 digital I/O pins
 - To connect devices that read/produce digital values (switch, leds, ...)
- Analog pins use many states using quantization
 - Arduino UNO has 6 input analog pins
 - To read analog sensors (temperature, pressure, ...)
 - An A/D converter of 10 bit returns integers from 0 to 1023

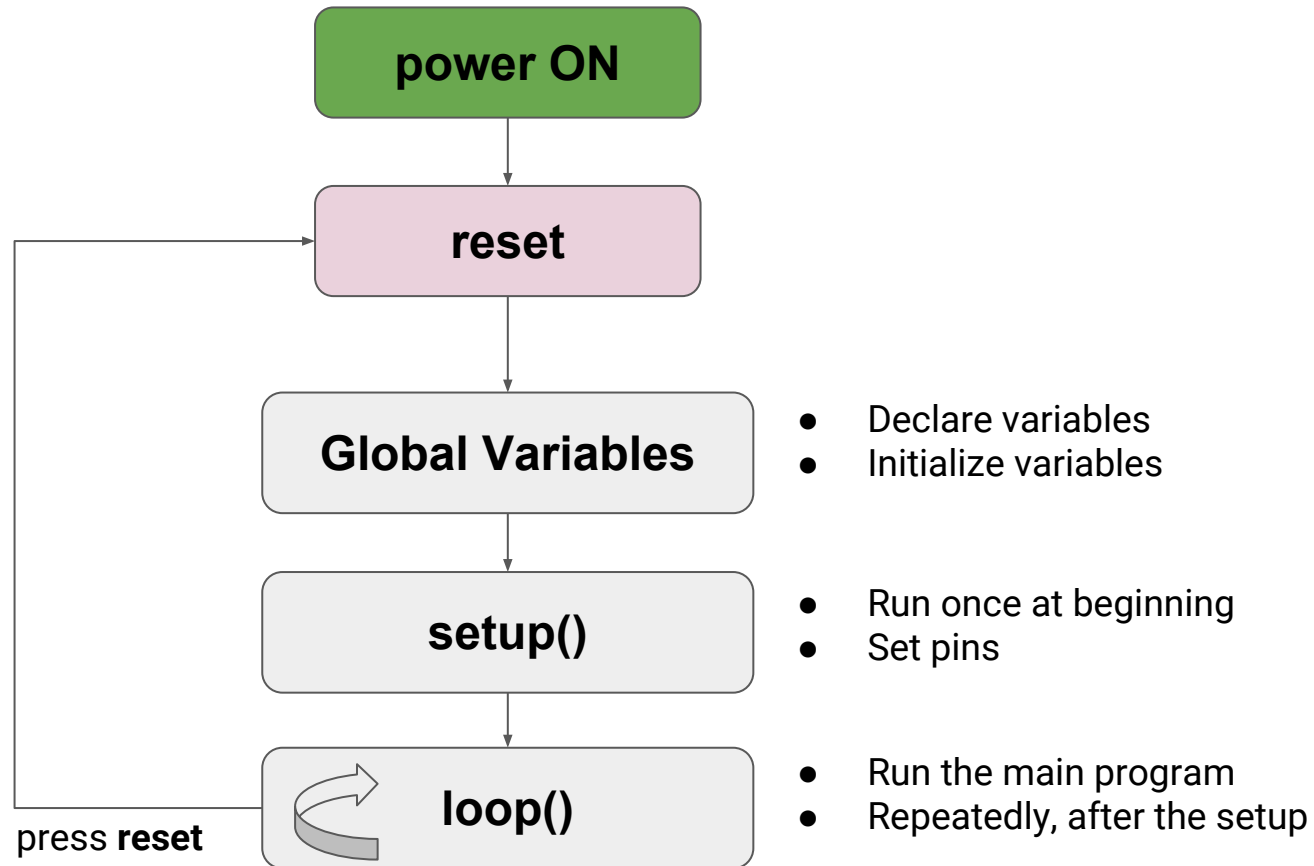


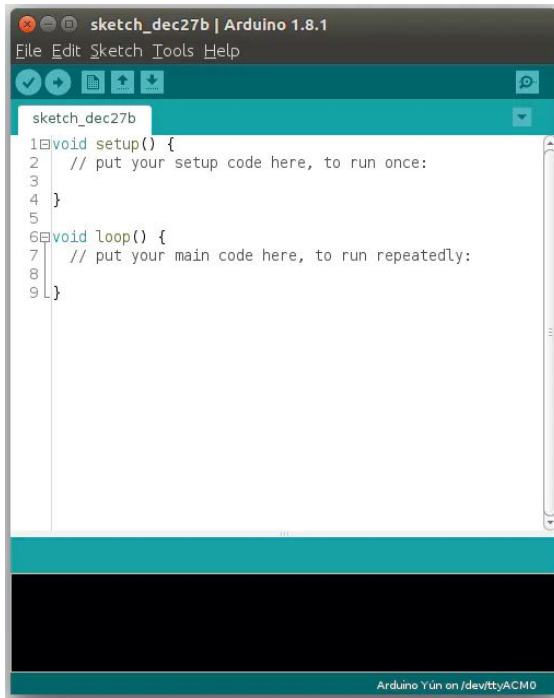
- Analog output pins use PWM technique
 - Arduino UNO has 6 digital pins used as PWM (~)
 - The signal is affected on the input pulse duration



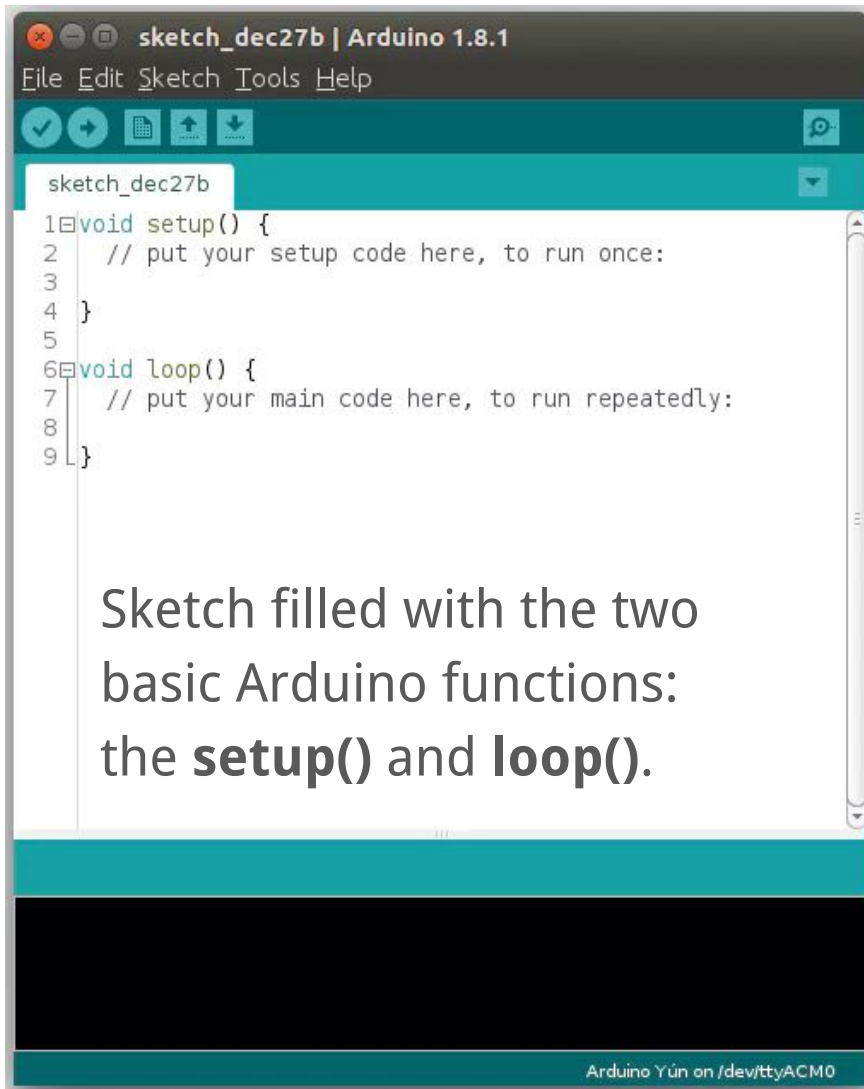
Programming in Arduino

Life Cycle of a program

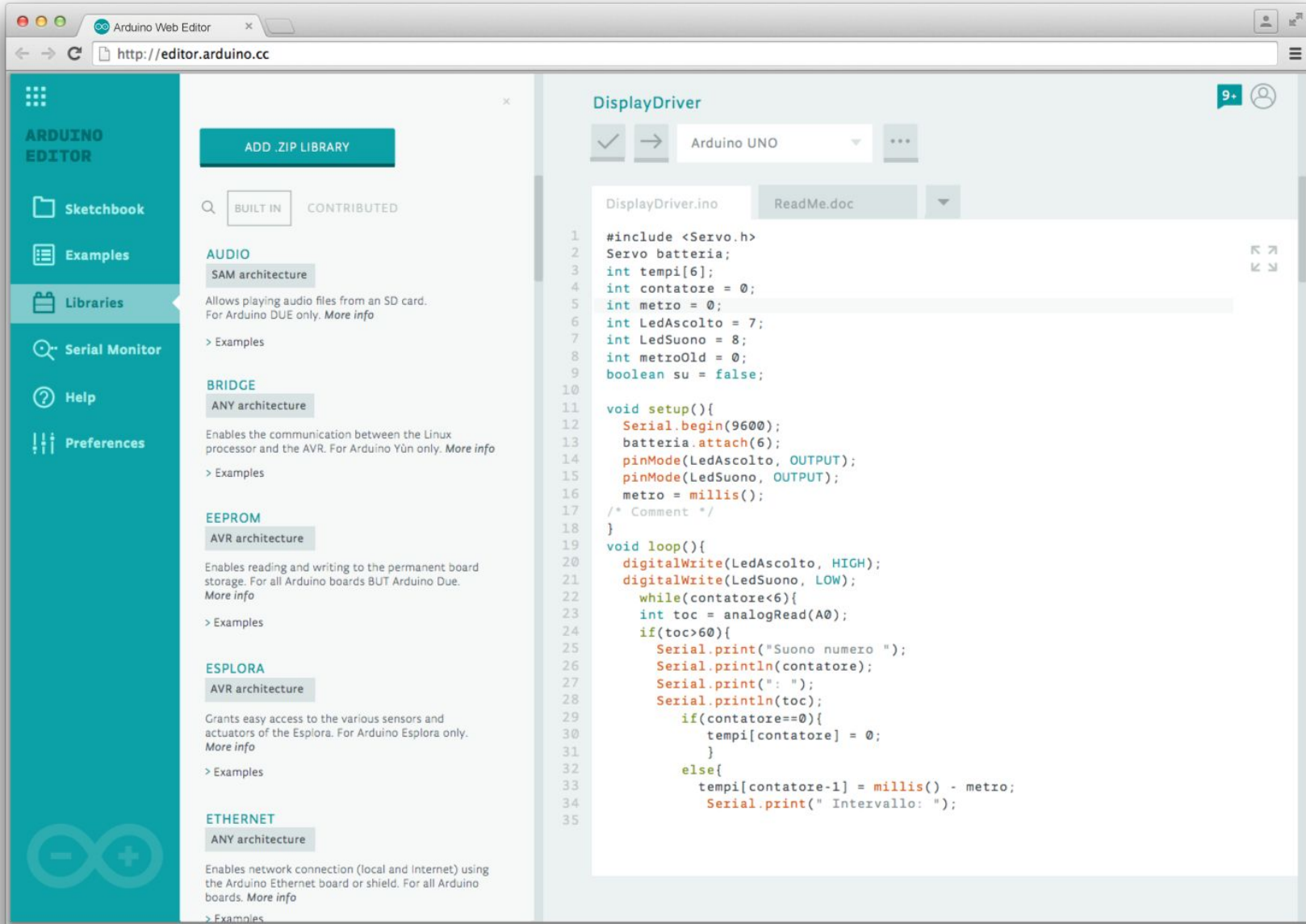




- Friendly Multi Platform application written in Java
- Allows you to write programs and upload them to your Arduino board
- Derived by "Processing" IDE
 - Initially developed in 2003 to simplify electronic projects
- Very simple for novice programmers
 - Simple programming language
 - Bootloader
- Source code of IDE and library are distributed under the free GNU GPLv2 license



- **Text editor** containing syntax highlighting and automatic indentation
- **Toolbar**
- **Text Console**
- **Compiler**
- **Serial monitor** to debug
 - Allows you to read the data that Arduino communicates through the COM serial port

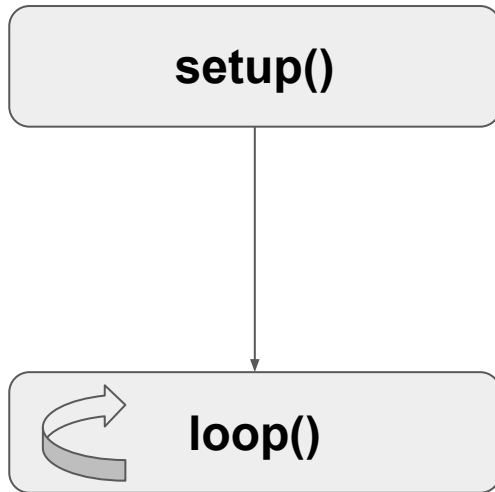


The screenshot displays the Arduino Web Editor interface in a web browser. The browser's address bar shows the URL `http://editor.arduino.cc`. The interface is divided into three main sections:

- Left Sidebar (Teal):** Contains navigation links: "Sketchbook", "Examples", "Libraries" (selected), "Serial Monitor", "Help", and "Preferences".
- Libraries Panel (Middle):** Shows a list of available libraries with search filters "BUILT IN" and "CONTRIBUTED". The "AUDIO" library is expanded, showing "SAM architecture" with a description: "Allows playing audio files from an SD card. For Arduino DUE only. [More info](#)". Other libraries like "BRIDGE", "EEPROM", "ESPLORA", and "ETHERNET" are also listed.
- Main Editor Area:** Displays a code file named "DisplayDriver.ino" for an "Arduino UNO" board. The code includes a servo motor and a timer, with functions for setup and loop. The code is as follows:

```
1 #include <Servo.h>
2 Servo batteria;
3 int tempi[6];
4 int contatore = 0;
5 int metro = 0;
6 int LedAscolto = 7;
7 int LedSuono = 8;
8 int metroOld = 0;
9 boolean su = false;
10
11 void setup(){
12   Serial.begin(9600);
13   batteria.attach(6);
14   pinMode(LedAscolto, OUTPUT);
15   pinMode(LedSuono, OUTPUT);
16   metro = millis();
17   /* Comment */
18 }
19 void loop(){
20   digitalWrite(LedAscolto, HIGH);
21   digitalWrite(LedSuono, LOW);
22   while(contatore<6){
23     int toc = analogRead(A0);
24     if(toc>60){
25       Serial.print("Suono numero ");
26       Serial.println(contatore);
27       Serial.print(": ");
28       Serial.println(toc);
29       if(contatore==0){
30         tempi[contatore] = 0;
31       }
32       else{
33         tempi[contatore-1] = millis() - metro;
34         Serial.print(" Intervallo: ");
35       }
36     }
37   }
38 }
```


- Open Source computer programming language
- Derived by C/C++ language
 - With some slight simplifications and modifications
 - Includes classical libraries and functions
 - Data types (Integer, float, long, character, ...)
 - Operators (Mathematical, logical, comparison, ...)
 - Control statements (If, switch/case, while, for, ...)
- Offer to the programmer simple access to I/O devices
- Wiring programs are called ***sketch***



- `pinMode(pin, Input|Output)` set pin ledPin as an input or output
- `Serial.begin(9600)` talk to the computer at 9600 baud rate
 - Some values: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200
- `Serial.print(" ... ")` write text on Serial Monitor
- `digitalWrite(pin, HIGH|LOW)` set a digital pin high/low
- `digitalRead(pin)` read a digital pin's state
- `analogRead(pin)` read an a analog pin
- `analogWrite(pin, intValue)` write an "analog" PWM value
- `delay(milliseconds)` wait an amount of time

Arduino's Hello World: LED blinking

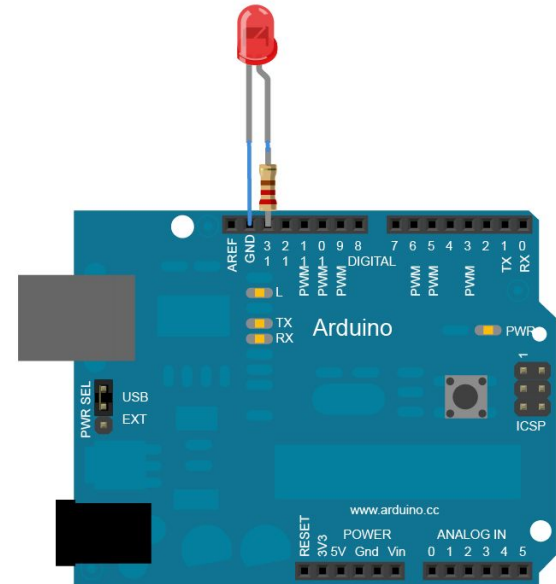


```
/*
  Blink
  Turn on and off a LED every one second
*/

int ledPin = 13; // LED connected to digital pin 13

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

// the loop routine runs over and over again forever:
void loop()
{
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                // wait for a second
}
```



Arduino's Hello World: LED blinking



```
/*  
  Blink  
  Turn on and off a LED every one second  
*/
```

```
int ledPin = 13; // LED connected to digital pin 13
```

Initialise the variable with the pin number

```
void setup()  
{  
  Serial.begin(9600);  
  pinMode(ledPin, OUTPUT); // sets the digital pin as output  
}  
  
// the loop routine runs over and over again forever:  
void loop()  
{  
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);                // wait for a second  
  digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);                // wait for a second  
}
```


Arduino's Hello World: LED blinking



```
/*
  Blink
  Turn on and off a LED every one second
*/

int ledPin = 13; // LED connected to digital pin 13

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

// the loop routine runs over and over again forever:
void loop()
{
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                // wait for a second
}
```

Setup the serial
connections and LED

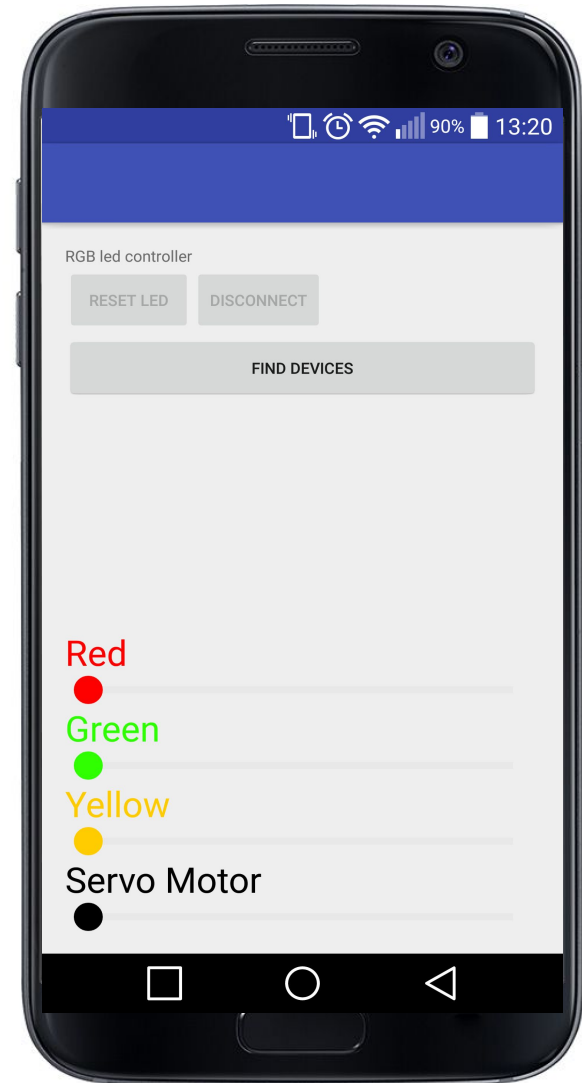
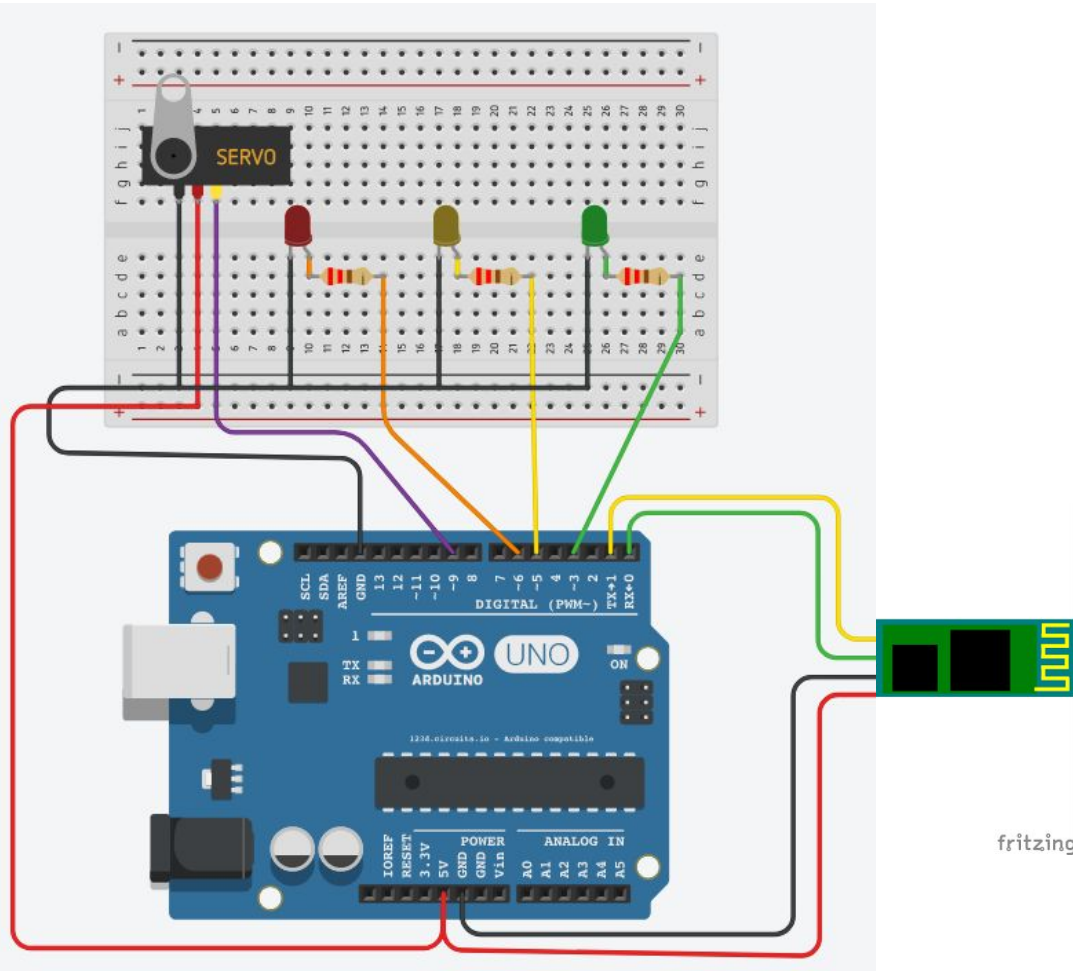
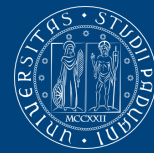
Arduino's Hello World: LED blinking



```
/*  
  Blink  
  Turn on and off a LED every one second  
*/  
  
int ledPin = 13; // LED connected to digital pin 13  
  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(ledPin, OUTPUT); // sets the digital pin as output  
}  
  
// the loop routine runs over and over again forever:  
void loop()  
{  
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);                // wait for a second  
  digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);                // wait for a second  
}
```

Turn the LED on and off
continuously in the loop

LED and Servo control via Android Application



Thank you for the attention

Daniele Ronzani

dronzani@math.unipd.it