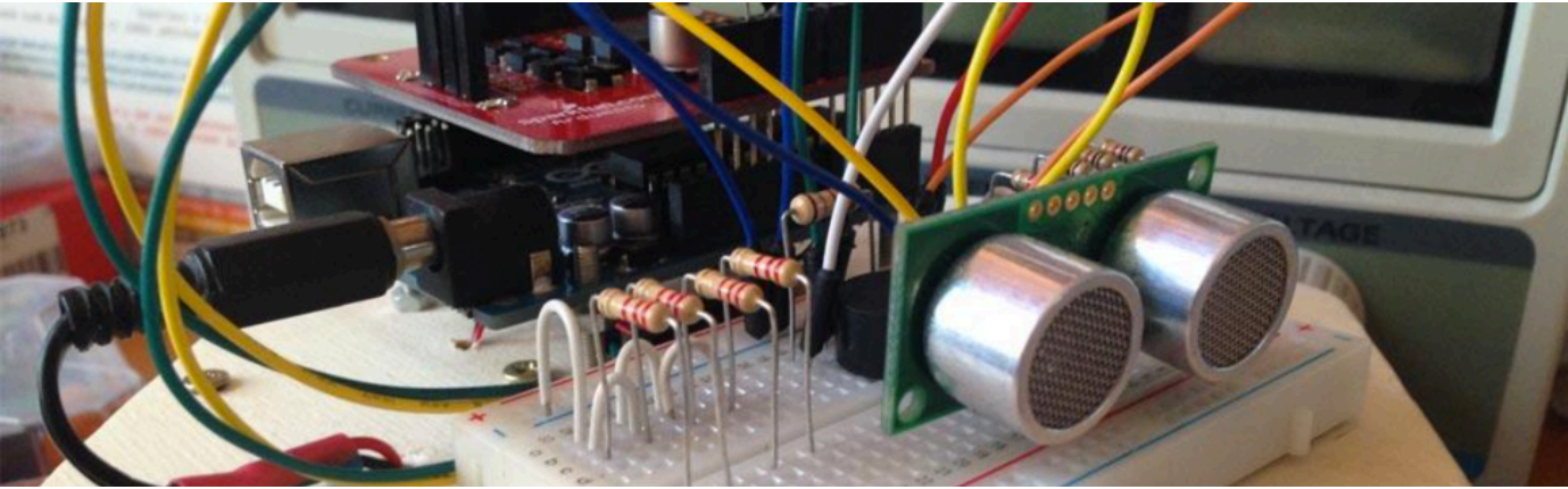
Electronic Prototyping

Introduction to Arduino use

Lesson 2

PhD Student Licia Di Pietro



Outline

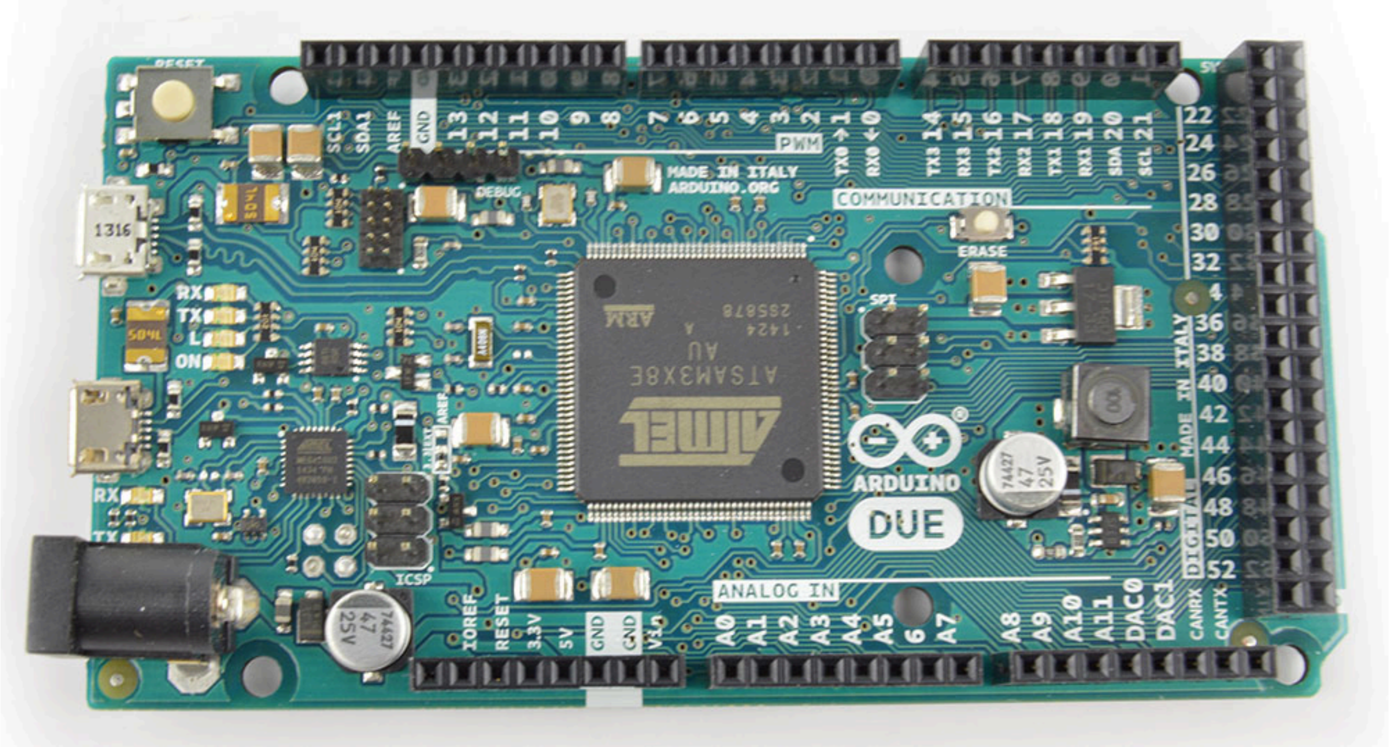
- **What is Arduino?**
- **Arduino hardware**
- **Arduino DUE Pin mapping**
- **Terminology**
- **The software**

What is Arduino? (1/4)

Arduino’s Word means **3 things**

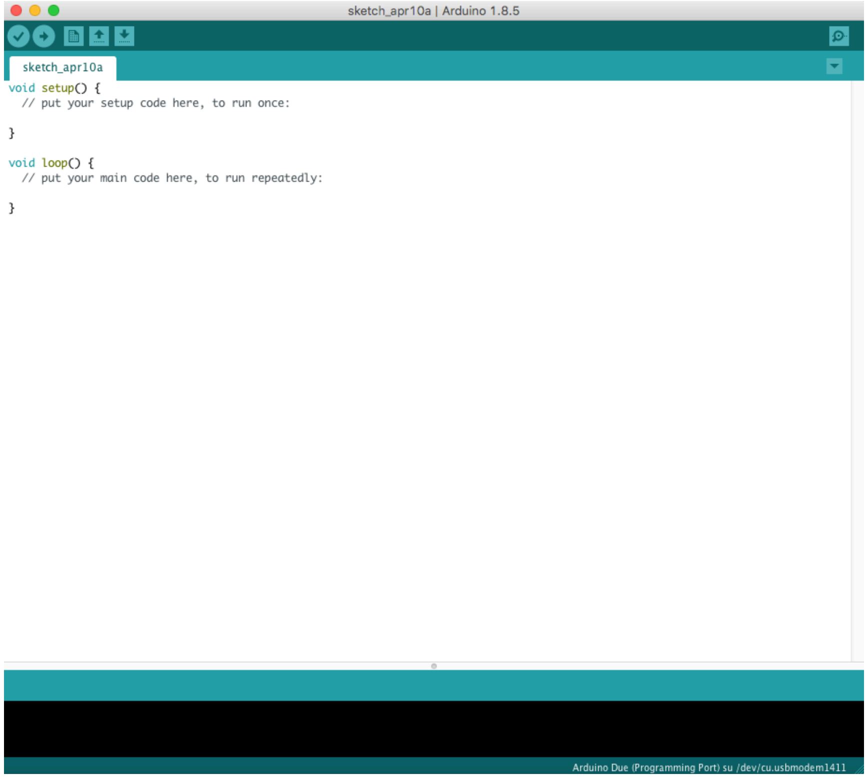
What is Arduino? (2/4)

- **Physical Object**



What is Arduino? (3/4)

- **Integrated Development Environment**



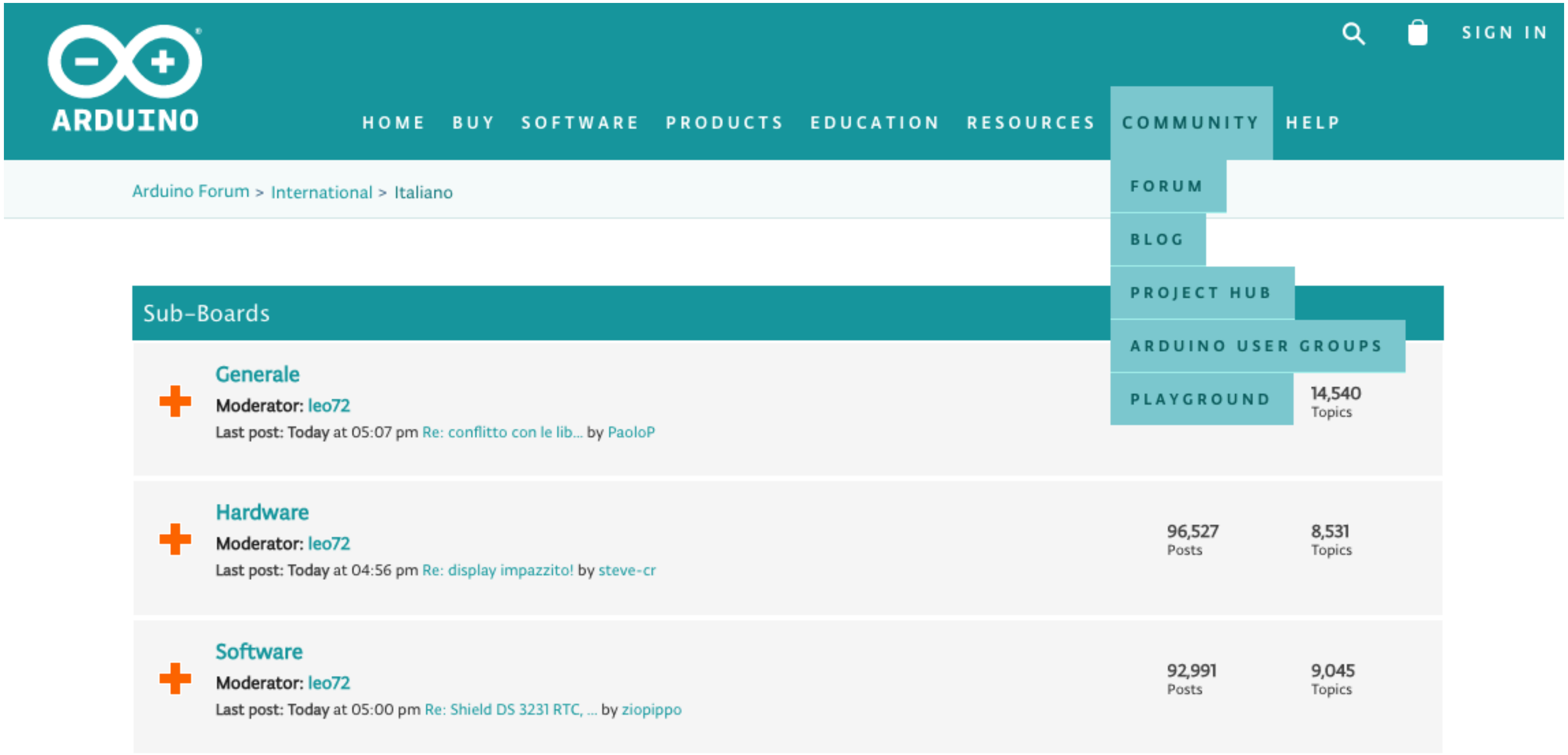
sketch_apr10a | Arduino 1.8.5
sketch_apr10a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Arduino Due (Programming Port) su /dev/cu.usbmodem1411

What is Arduino? (4/4)

- **A community and a development philosophy**



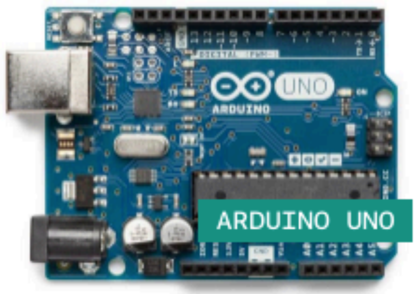
Arduino Forum > International > Italiano

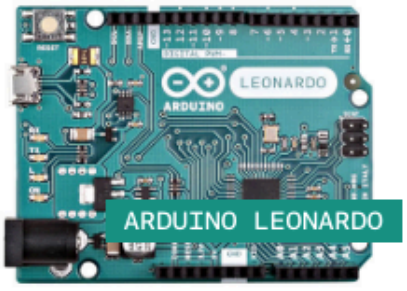
Sub-Boards

Generale Moderator: leo72 Last post: Today at 05:07 pm Re: conflitto con le lib... by PaoloP		14,540 Topics
Hardware Moderator: leo72 Last post: Today at 04:56 pm Re: display impazzito! by steve-cr	96,527 Posts	8,531 Topics
Software Moderator: leo72 Last post: Today at 05:00 pm Re: Shield DS 3231 RTC, ... by ziopippo	92,991 Posts	9,045 Topics

Arduino Hardware (1/5)

- **Entry Level**


ARDUINO UNO


ARDUINO LEONARDO

ARDUINO 101

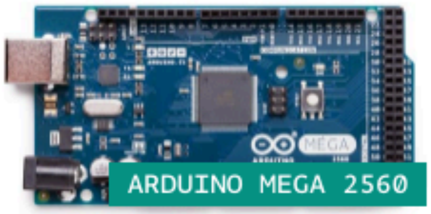
ARDUINO ESPLORA

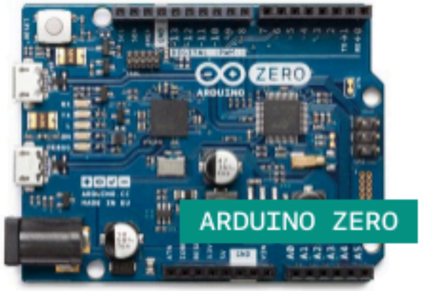
ARDUINO MICRO

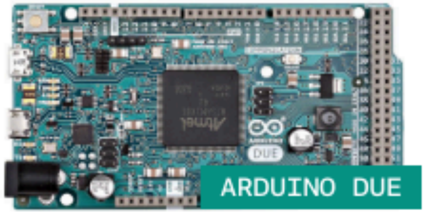
ARDUINO NANO

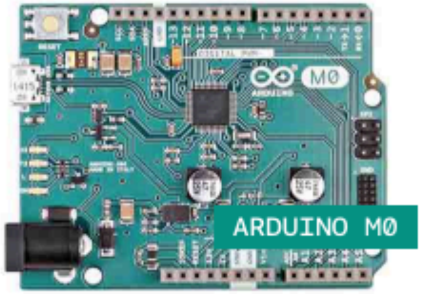
Arduino Hardware (2/5)

- **Enhanced Features**



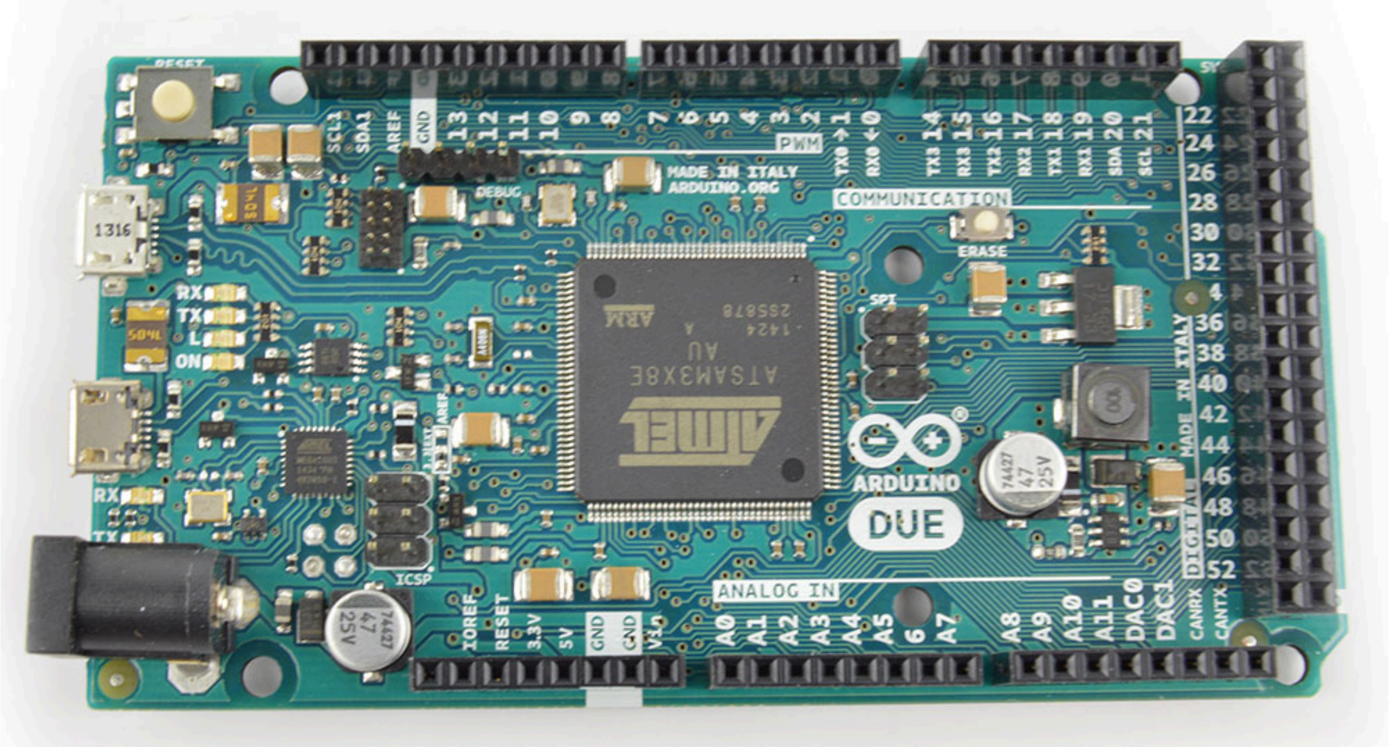






Arduino Hardware (3/5)

- **In this course we will use Arduino DUE**



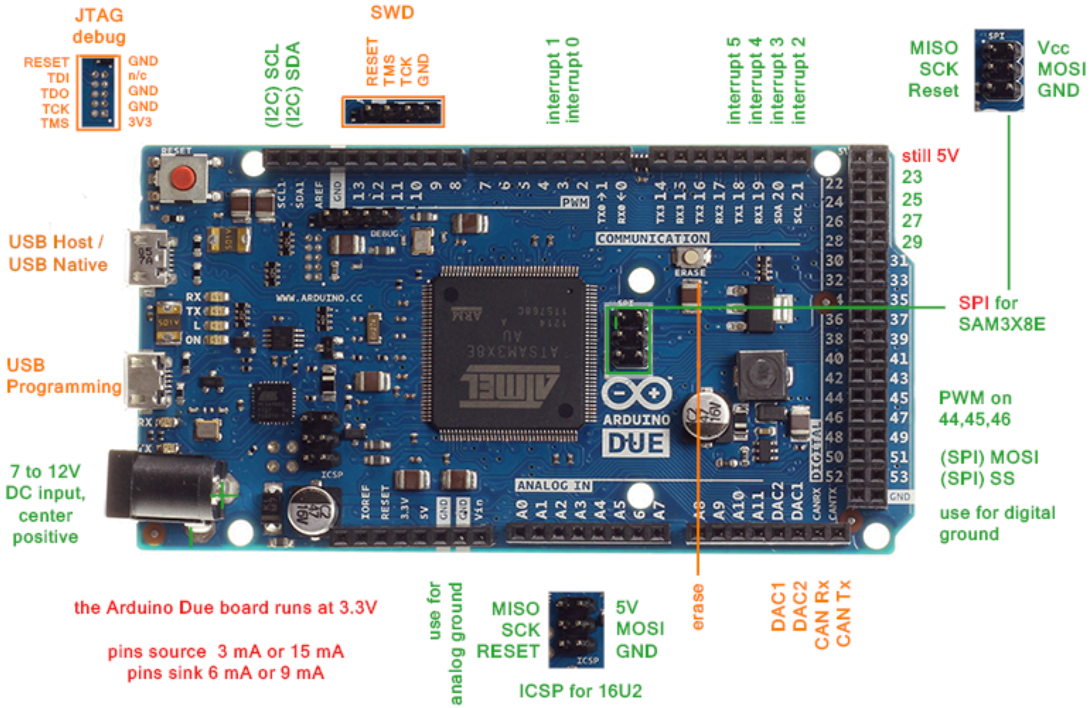
Arduino Hardware (4/5)

- **Technical Specification**

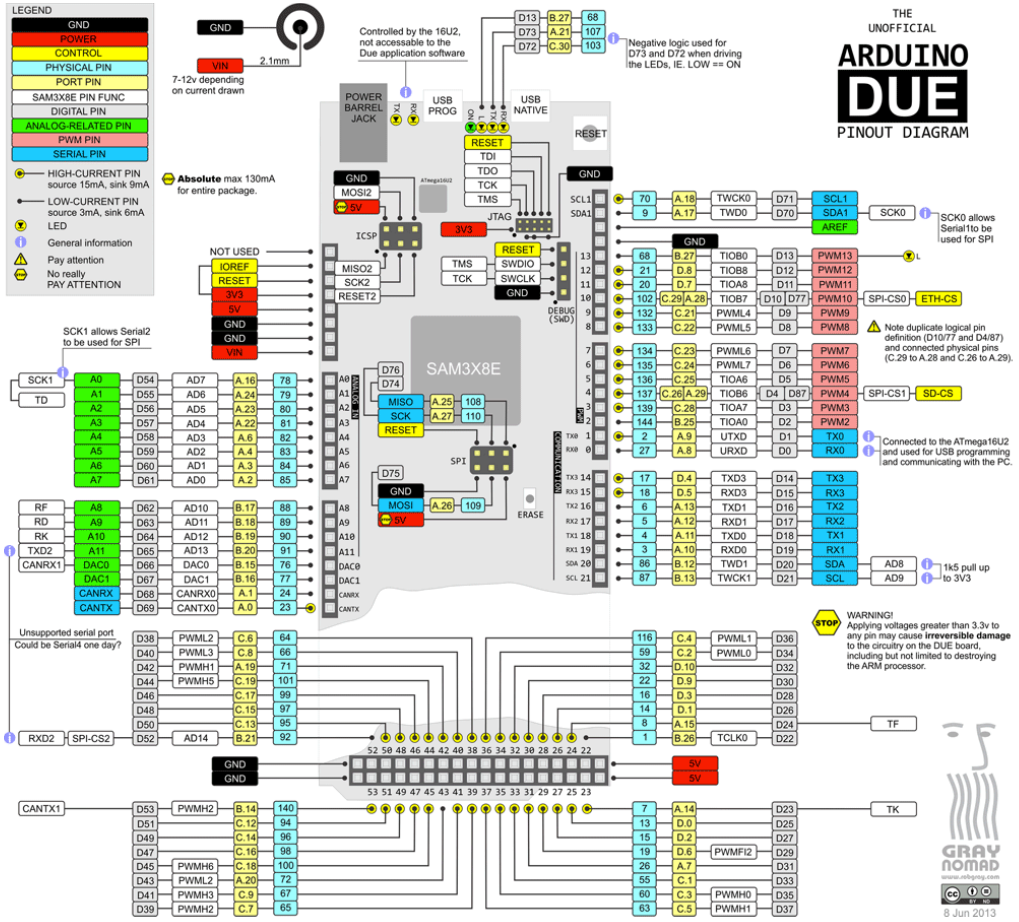
Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

Arduino Hardware (5/5)

- **Fundamental Parts**



Arduino DUE Pin Mapping



7-12v depending on current drawn

Controlled by the 16U2, not accessable to the Due application software

Negative logic used for D73 and D72 when driving the LEDs, IE. LOW == ON

Absolute max 130mA for entire package.

SCK1 allows Serial2 to be used for SPI

SCK0 allows Serial1to be used for SPI

Note duplicate logical pin definition (D10/77 and D4/87) and connected physical pins (C.29 to A.28 and C.26 to A.29).

Connected to the ATmega16U2 and used for USB programming and communicating with the PC.

1k5 pull up to 3V3

Unsupported serial port Could be Serial4 one day?

WARNING! Applying voltages greater than 3.3v to any pin may cause **irreversible damage** to the circuitry on the DUE board, including but not limited to destroying the ARM processor.

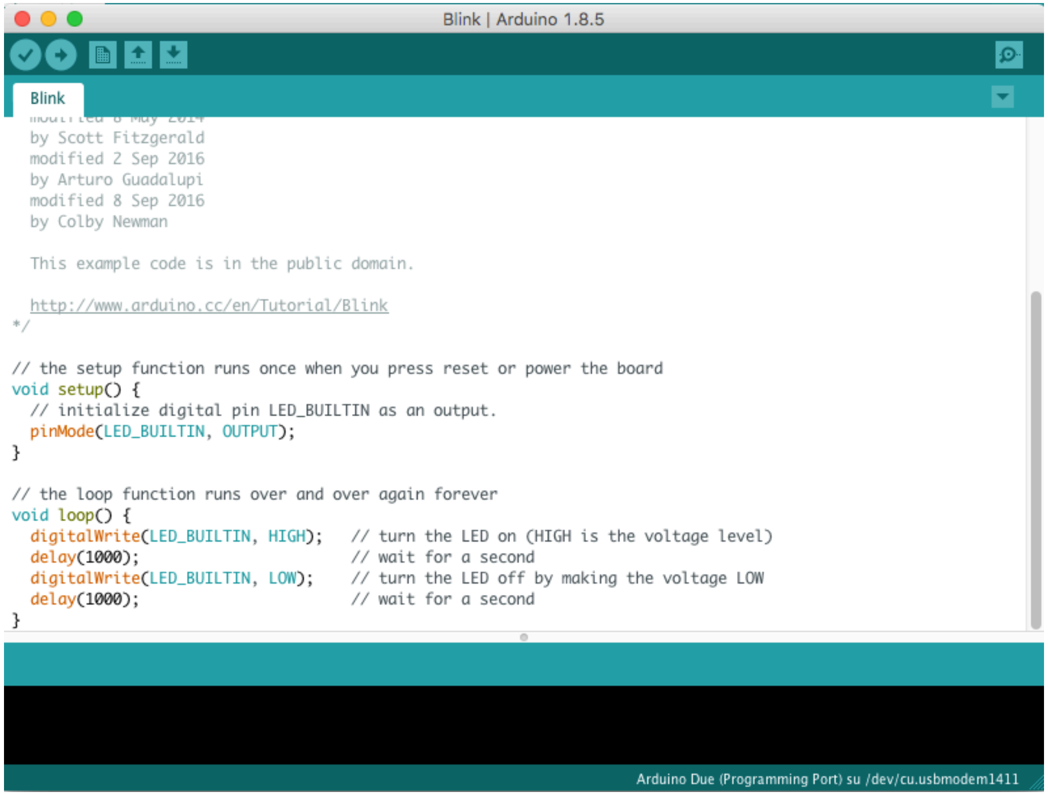
8 Jun 2013

Terminology

- **sketch**
  - Is the program that you write and run on the Arduino board
- **pin**
  - Input and output connectors
- **digital**
  - It means that it can only take two values: HIGH or LOW, in another way ON/OFF or 0/1
- **analog**
  - When the values are continuous (infinite)

The Software (1/2)

- Similar to a text editor;
- You can write, visualize and verify the syntax;
- You can upload the sketch on your board.



```
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

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

The Software (2/2)

CLARIFICATION

During these lessons to indicate the software development environment, we will use:

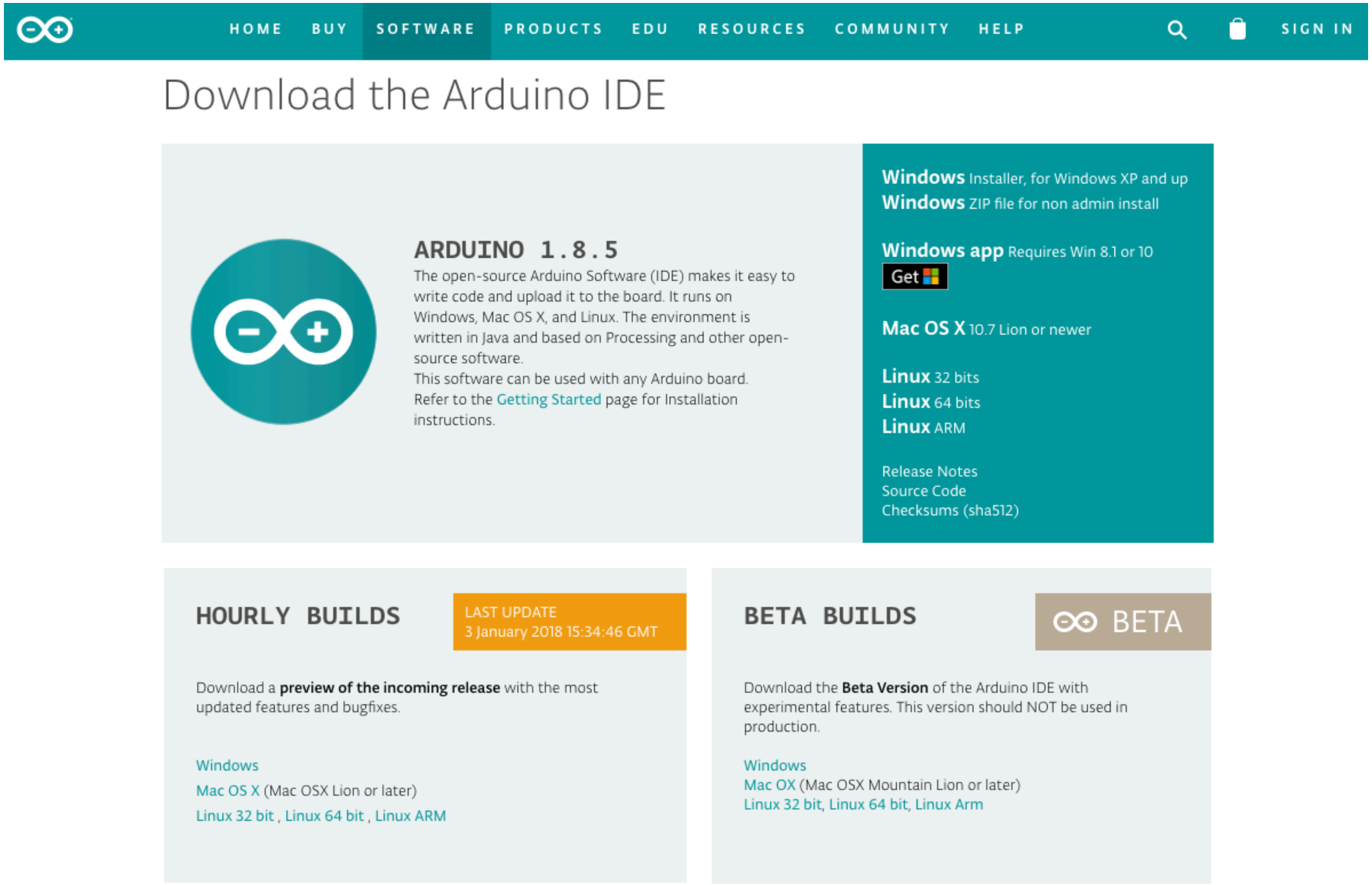
software Arduino

Or with the same means:

IDE

IDE means: *Integrated Development Enviroment*, in italiano: *ambiente di sviluppo integrato per la realizzazione di programmi.*

Software Installation



Download the Arduino IDE

ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

HOURLY BUILDS

LAST UPDATE
3 January 2018 15:34:46 GMT

Download a **preview of the incoming release** with the most updated features and bugfixes.

Windows
Mac OS X (Mac OSX Lion or later)
Linux 32 bit , Linux 64 bit , Linux ARM

BETA BUILDS

BETA

Download the **Beta Version** of the Arduino IDE with experimental features. This version should NOT be used in production.

Windows
Mac OX (Mac OSX Mountain Lion or later)
Linux 32 bit, Linux 64 bit, Linux Arm

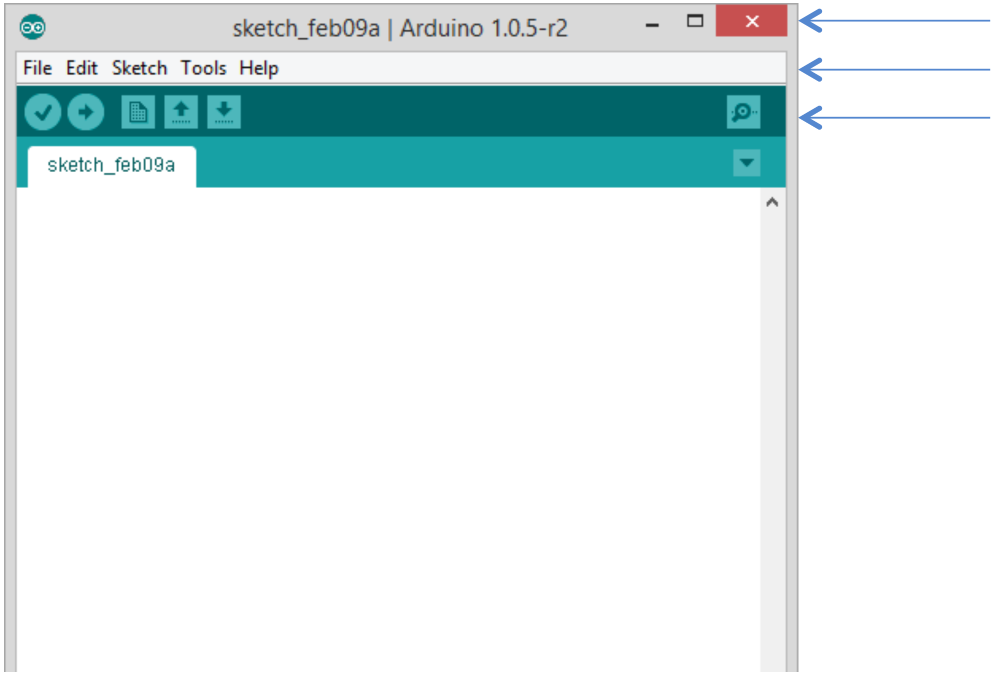
Communication with Arduino

- **Launch the Arduino IDE (double click)**

Arduino Program Development

- Based on C++ without 80% of the instructions.
- A handful of new commands.
- Programs are called 'sketches'.
- Sketches need two functions:
  - void setup( )
  - void loop( )
- setup( ) runs first and once.
- loop( ) runs over and over, until power is lost or a new sketch is loaded.

Parts of the IDE main screen



Name of current sketch

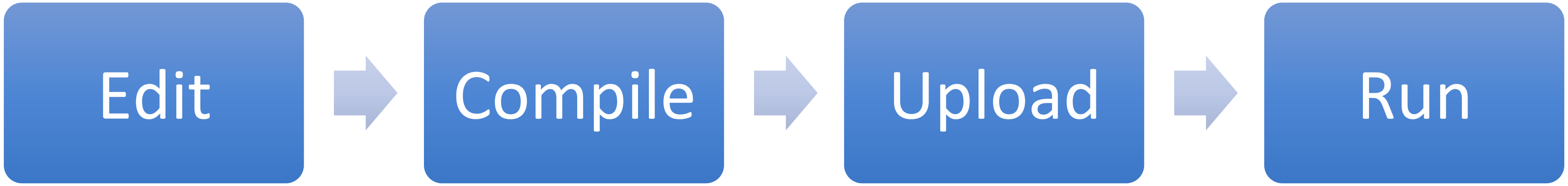
Main menus

Action buttons/icons

- Verify (AKA compile)
- Upload (send to Arduino)
- Start a new sketch
- Open a sketch (from a file)
- Save current sketch (to a file)
- Open Serial Monitor window

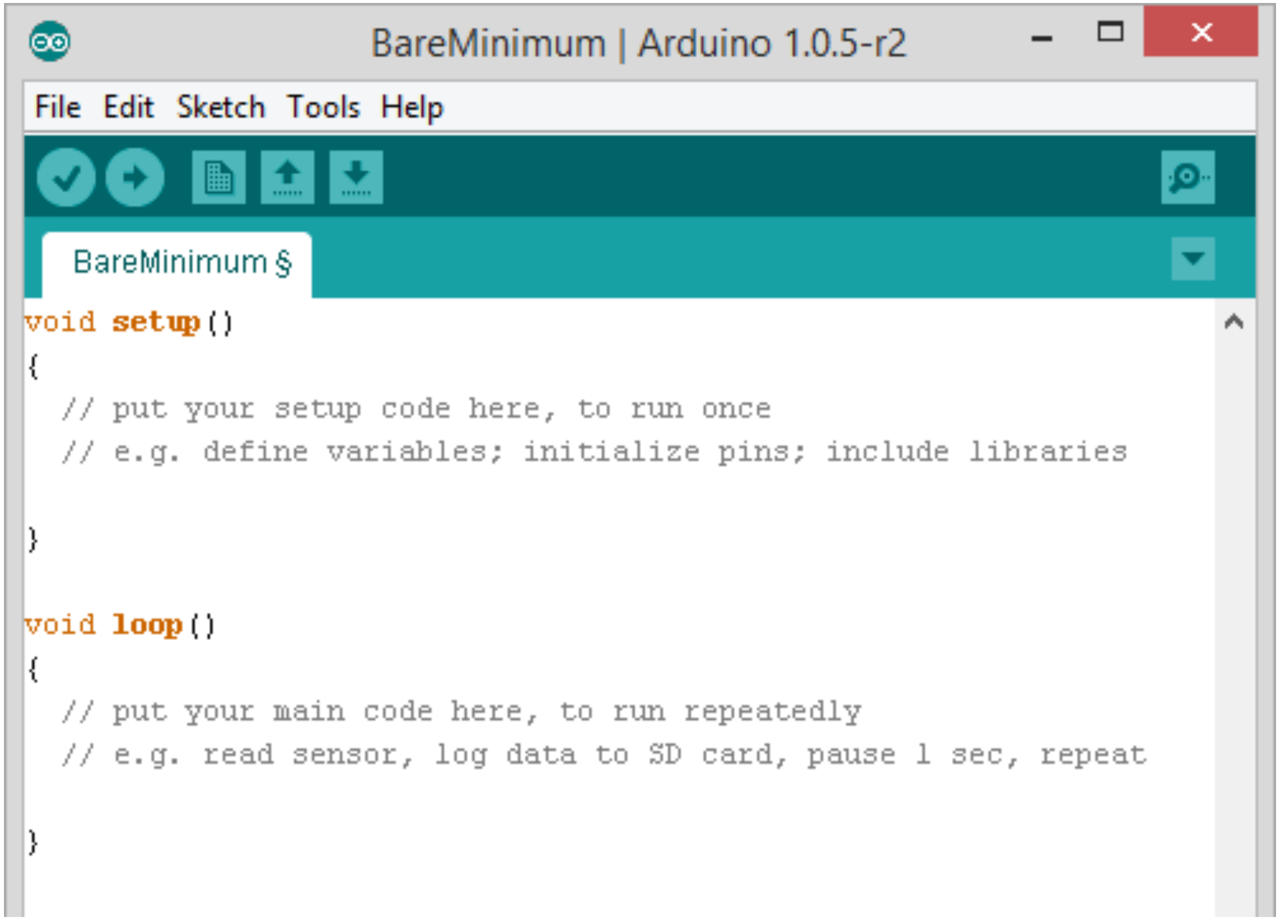
Programming

Development Cycle



Compile: Compile means to translate the sketch into machine language, also known as object code
Run: Arduino sketch is executed as soon as terminates the step of uploading on the board

The structure of Arduino Sketch (1/2)



```
void setup()
{
  // put your setup code here, to run once
  // e.g. define variables; initialize pins; include libraries

}

void loop()
{
  // put your main code here, to run repeatedly
  // e.g. read sensor, log data to SD card, pause 1 sec, repeat

}
```

The structure of Arduino Sketch (2/2)

- The first one is “**setup()**”. Anything you put in this function will be executed by the Arduino just once when the program starts.

- The second one is “**loop()**”. Once the Arduino finishes with the code in the **setup()**function, it will move into **loop()**, and it will continue running it in a loop, again and again, until you reset it or cut off the power.

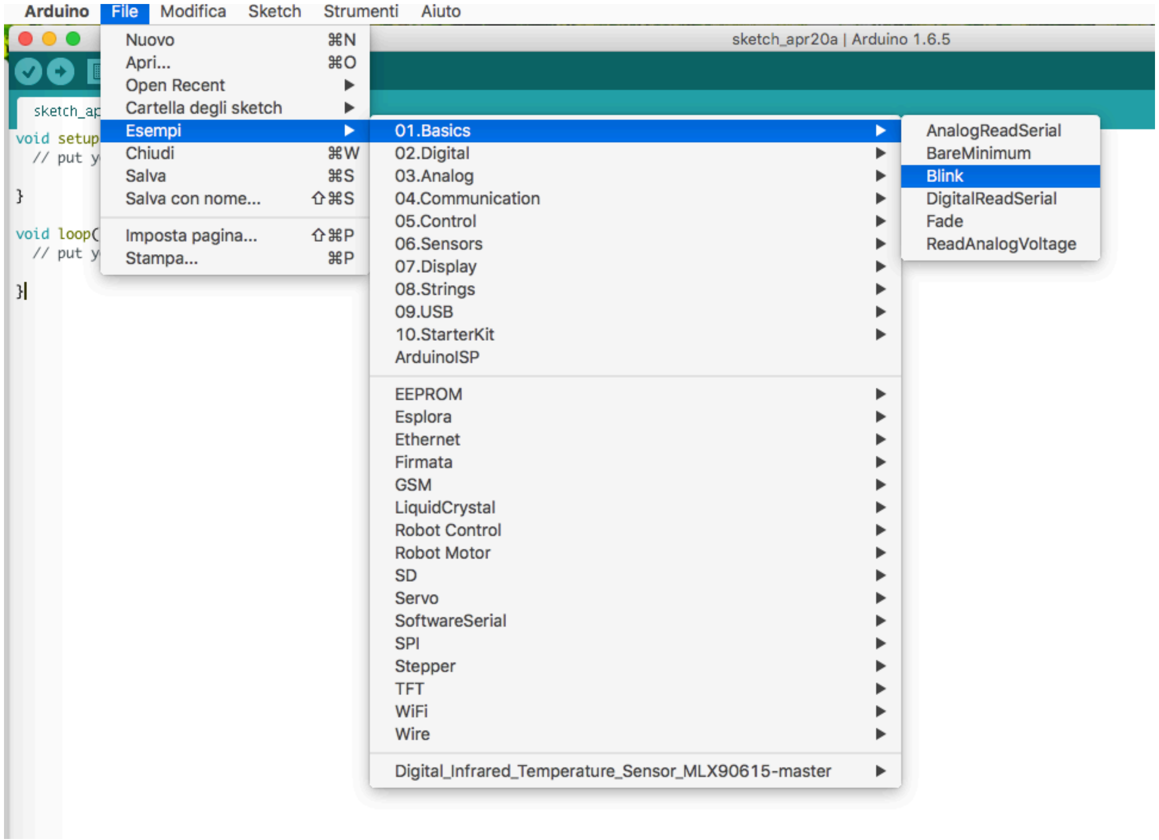
Arduino specific function

- **pinMode(*pin*, *mode*)**
  - Configures a digital pin to read (input) or write (output) a digital value
- **digitalWrite(*pin*, *value*)**
  - Writes the digital value (HIGH or LOW) to a pin set for output
- **digitalRead(*pin*)**
  - Reads a digital value (HIGH or LOW) on a pin set for input
- **analog versions of above**
  - **analogRead's** range is 0 to 1023 (for Arduino Uno)
  - The Due and the Zero have 12-bit ADC capabilities that can be accessed by changing the resolution to 12. This will return values from analogRead() between 0 and 4095.
- **serial commands**
  - print, println, write, delay
- Other example

  https://www.arduino.cc/en/Reference/HomePage

Arduino Sketch Example

- Numerous sample sketches are included in the compiler
- Located under File, Examples


Arduino
File
Modifica
Sketch
Strumenti
Aiuto
sketch_apr20a | Arduino 1.6.5
sketch_ap
void setup
// put y
}
void loop(
// put y
}
Nuovo ⌘N
Apri... ⌘O
Open Recent ▶
Cartella degli sketch ▶
Esempi ▶
Chiudi ⌘W
Salva ⌘S
Salva con nome... ⇧⌘S
Imposta pagina... ⇧⌘P
Stampa... ⌘P
01.Basics ▶
02.Digital ▶
03.Analog ▶
04.Communication ▶
05.Control ▶
06.Sensors ▶
07.Display ▶
08.Strings ▶
09.USB ▶
10.StarterKit ▶
ArduinoISP
EEPROM ▶
Esplora ▶
Ethernet ▶
Firmata ▶
GSM ▶
LiquidCrystal ▶
Robot Control ▶
Robot Motor ▶
SD ▶
Servo ▶
SoftwareSerial ▶
SPI ▶
Stepper ▶
TFT ▶
WiFi ▶
Wire ▶
Digital_Infrared_Temperature_Sensor_MLX90615-master ▶
AnalogReadSerial
BareMinimum
Blink
DigitalReadSerial
Fade
ReadAnalogVoltage