ARDUINO DAYS 2024

Arduino: What is it?

By Marco Baruzzo

ICTP Scientific Fablab

ICTP SCI FAB LAB

scifablab.ictp.it

Why are we here today?

- Let you discover what Arduino is
- Let you discover what Arduino can do for you
- Give you some words of the microcontroller vocabulary

Why are we here today?

What are the words that we are going to add to our vocabulary?:

- Microcontroller
- Input – Output pins (I/O, Digital, Analog, PWM, GPIO)
- Protocols (I2C, SPI)
- Libraries
- Arduino Integrated Development Environment (IDE)
- C and C++ dialects and Sketch
- Shield

These words will recur throughout this presentation.

Let’s do a step back, why are we here today?

I want to know more about Arduino, which is like asking what can electronics do for me?

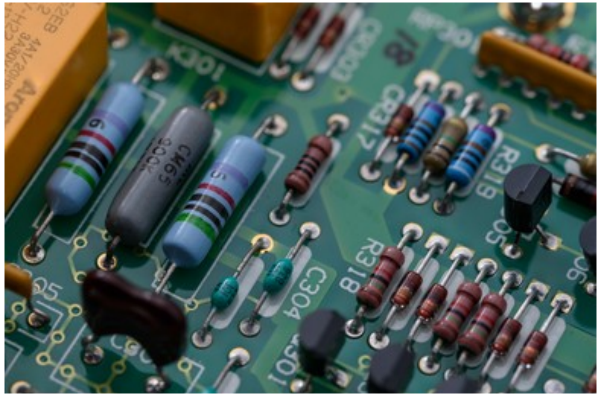
- Measure things
- Control things
- Do things faster
- Do boring things
- Do things automatically based on something measured
- …….

A lot of things=)

Why are we here today?

I want electronics to do things for me…

Most of the simpler things that we usually do can be made also by analog circuit, but this requires a lot of knowledge and components (e.g. thermostats)

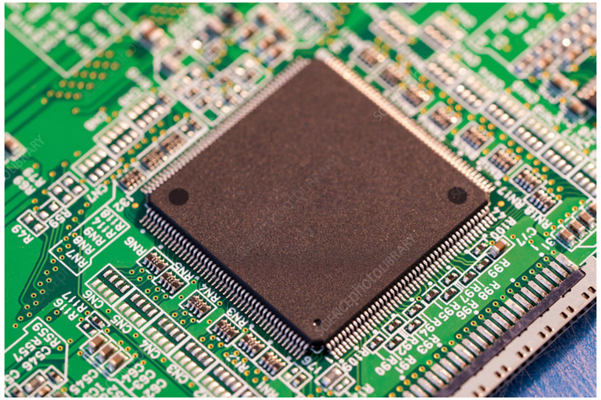


Analog/On purpose

Why are we here today?

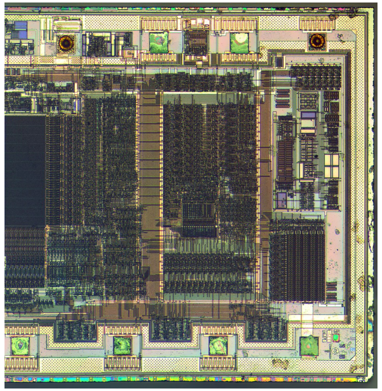
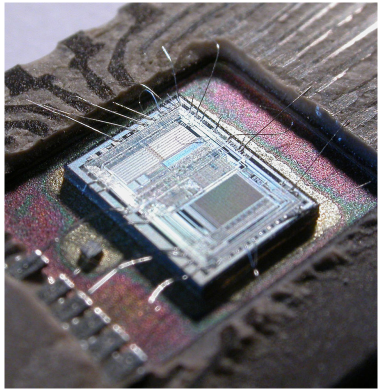
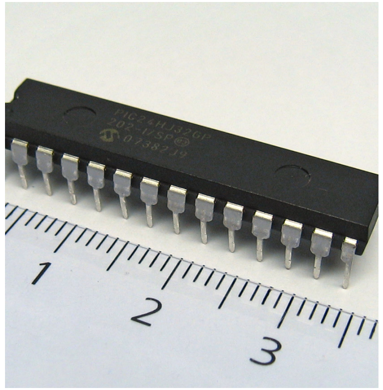
I want electronics to do things for me…

If you have basis of programming (or not) the easiest path can be the use of a microcontroller



Digital/**Programmable**

What is a microcontroller?

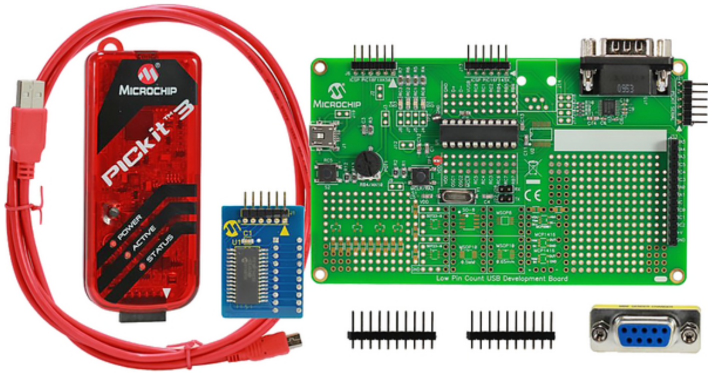




Integrated circuit based on semiconductor

This can be programmed to perform a specific task using a computer code called firmware, that you have to write

Before Arduino

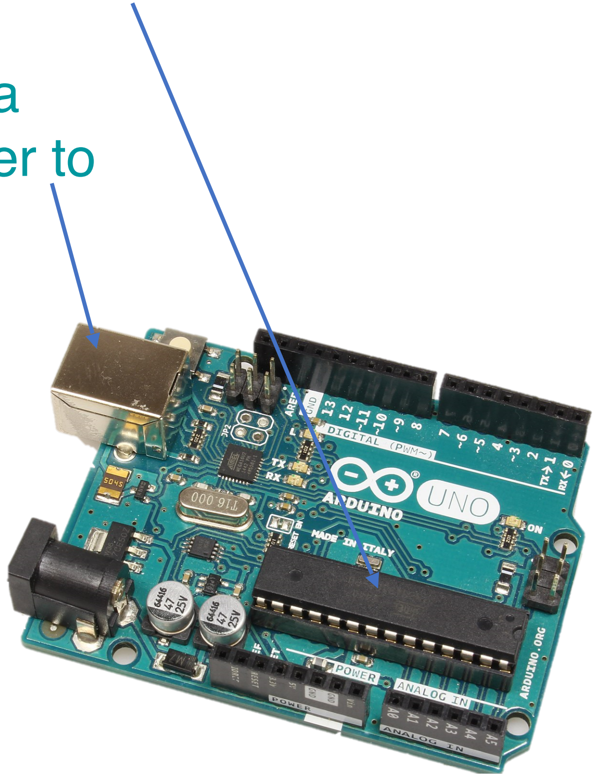
- Complex coding (programming for PLC) – Simpler coding
- Lack of software support (libraries) – More Libraries
- Scarcity of available (daughter)boards (you needed to develop your own) – A standard form factor to connect all the electronics that I want
- Use of electronics for R&D was "hard"



Arduino

It is an electronic board hosting a microcontroller.
This powerful combination enables you:

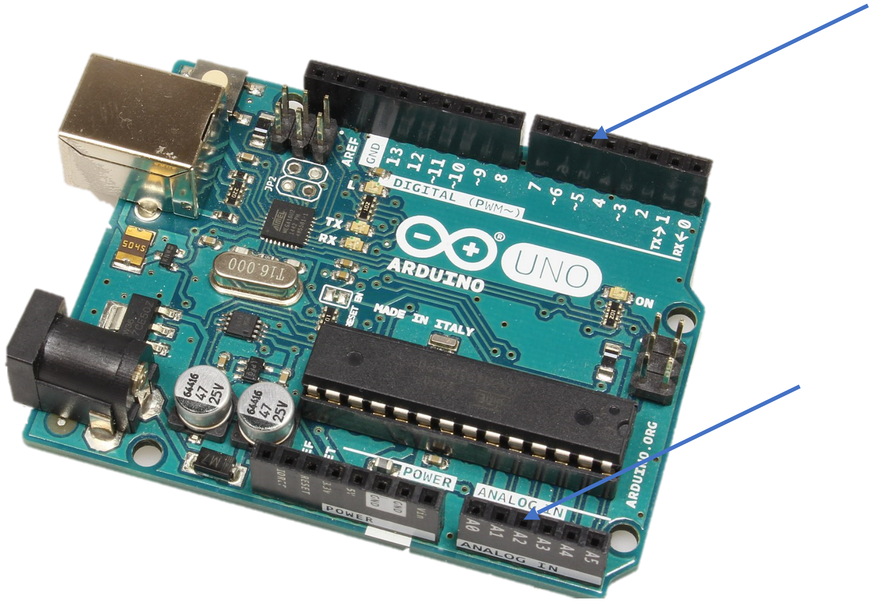
- To connect the microprocessor with the PC in a simple way, using serial-USB converter, in order to transfer the firmware
- To supply power to additional electronics
- To connect other stuff to the digital and analog pins of the microcontroller via jumper cables



I/O pins

An I/O pin is an interface between the microcontroller and the external world, these can be:

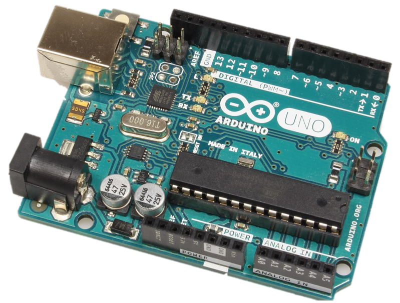
- Input
- Output
- Analog
- Digital
- PWM
- General Purpose



I/O pins

The inputs are the way in which the microcontroller acquire information and output are used by the microcontroller to “act” on the external environment

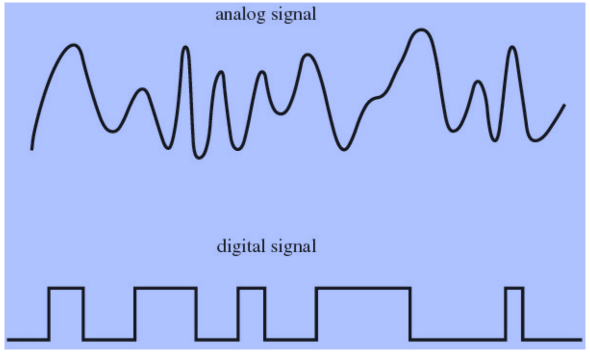
As human we use light and light sensors (eyes) to acquire most of the information (and other 4 senses)
The microcontroller uses voltage (electrical signals)



I/O pins

Voltage can vary continuously, typically from 0 to 5 Volts, to represent a continuous value (like the luminosity of a light source). This is called Analog value.

Or voltage can be set to one of two “levels”, High and Low, typically 5 Volts and 0 Volts, to say On / Off (there is light, there’s not light). This is called Digital value.



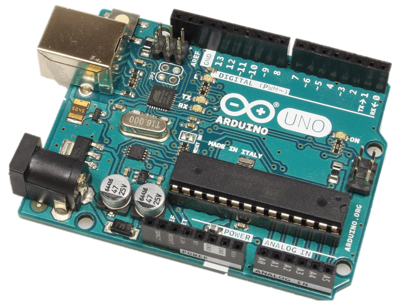
I/O pins

Most Arduinos have a set of analog inputs to measure voltages from sensors and a set of digital inputs working also as outputs, I/O.

Some digital output can also perform a Pulse-Width Modulation or PWM. This technique simulates an Analog Output.

Pins on most advanced microcontrollers can be used as analog or digital and are called General Purpose I/O, or GPIO

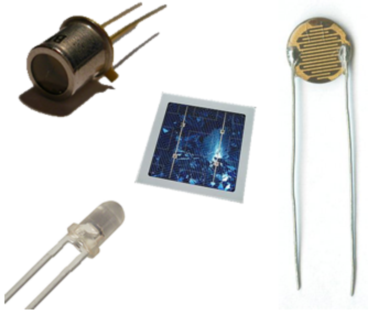
I'm tired to change color of words =), now you know what is a micro and pins!

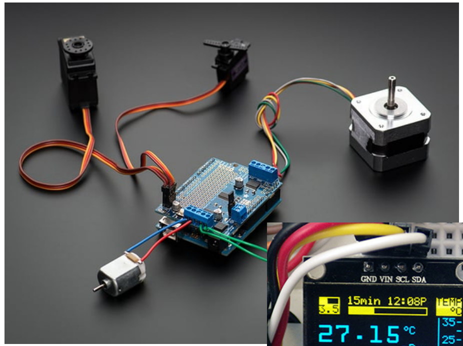


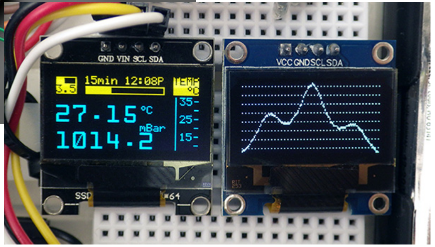
What can I connect to the I/O pins?

- LED =)
- A lot of different types of sensors
- Controllers (Buttons, Analog sticks, “touch”)
- Motors (DC motors, Steppers, Servos)
- Relays (controlled switches)
- Displays (LCD, OLED)
- Speakers (Not so simple)
- Memory (SD card)

And we can control them with code







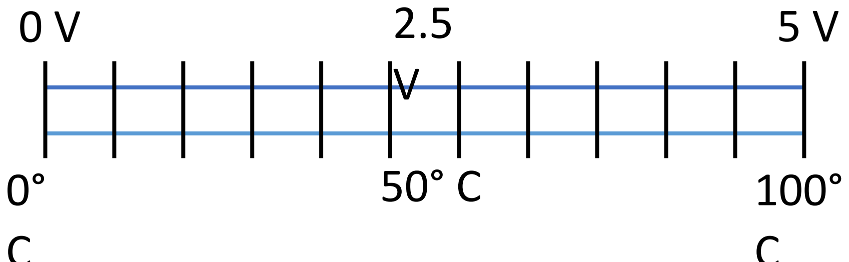
Sensors

- Temperature
- Humidity
- Pressure
- Light
- Strain
- Weight
- Accelerometer
- Movement
- Distance
- Presence
- Sound (microphone)
- ……

Most of these sensors can be analog or digital.

Analog means that the sensor will produce a voltage "proportional" to the value that you want to measure

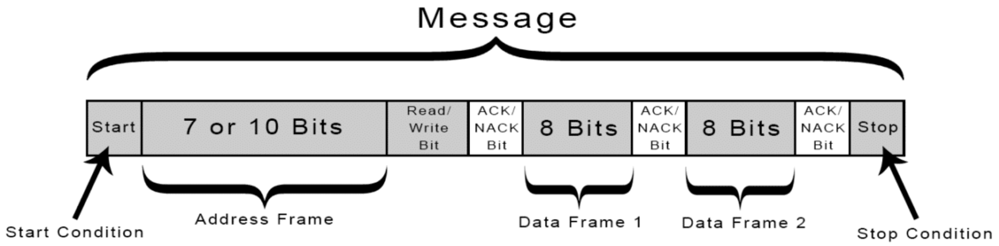
This means that you have to measure the voltage with an analog pin and do a calculation (mapping) to get the value of the desired quantity (i.e. temperature, etc). Defining the relationship between voltage and, i.e., temperature is called calibration.



Sensors

- Temperature
- Humidity
- Pressure
- Light
- Strain
- Weight
- Accelerometer
- Movement
- Distance
- Presence
- Sound (microphone)
- ……

Digital sensors have (internally) another small controller that measures and does all the math required to have the result, but you need to collect it with a digital pin of the Arduino.
And this is encoded in a series of 0 and 1 in a particular format that you need to decode.
This procedure is called a protocol



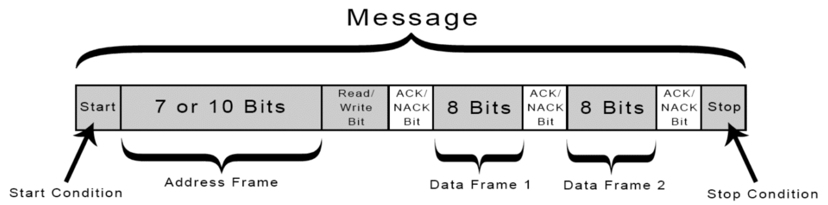
Protocols

A communication protocol is a standard set of rules that describes how two circuits speaks to each other.

Common Arduino examples are I2C, SPI, UART, OneWire

Other example? USB =)

Ok, Ok, Do I need to learn all this stuff?
Luckily not and this is why Arduino is so great



Library

Luckily someone did it for us =)

All these rules are coded into libraries, which are a set of simple commands that can be added to the standard programming language, to simplify the use of the devices.

A library can be downloaded and it usually comes with a set of instructions and examples that help you to use it.
Libraries are the reason why with Arduino is so simple to use. Together with a lot of guides online =)

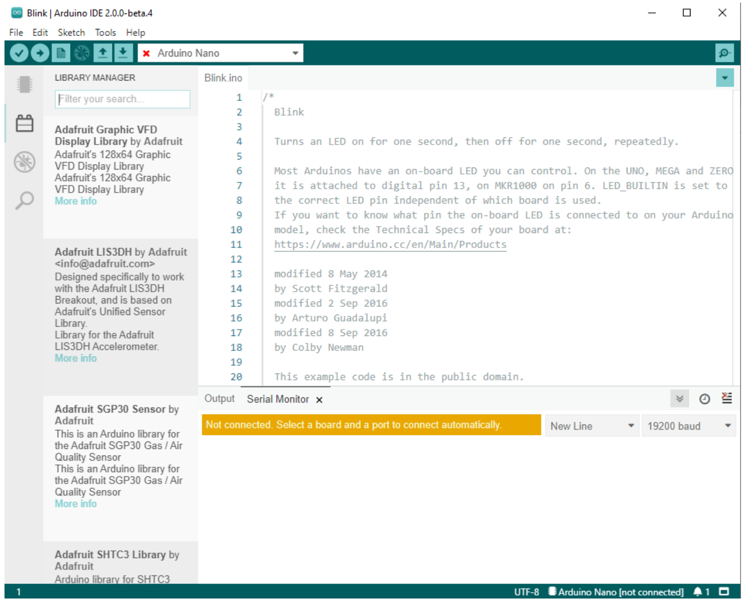
And with a friendly environment where to write the code…

Arduino IDE

… that is also the Arduino Integrated Development Environment (IDE).

This is a PC program where you can write the code. It should then be compiled (translated to "machine code") before it can run on the microcontroller.

You will also use the IDE to upload (transfer) the code on the microcontroller.



Adafruit Graphic VFD Display Library by Adafruit
Adafruit's 128x64 Graphic VFD Display Library
Adafruit's 128x64 Graphic VFD Display Library
More info

Adafruit LIS3DH by Adafruit <info@adafruit.com>
Designed specifically to work with the Adafruit LIS3DH Breakout, and is based on Adafruit's Unified Sensor Library.
Library for the Adafruit LIS3DH Accelerometer.
More info

Adafruit SGP30 Sensor by Adafruit
This is an Arduino library for the Adafruit SGP30 Gas / Air Quality Sensor
This is an Arduino library for the Adafruit SGP30 Gas / Air Quality Sensor
More info

Adafruit SHTC3 Library by Adafruit
Arduino library for SHTC3

```
/*
  Blink

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

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  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.
```

Not connected. Select a board and a port to connect automatically.

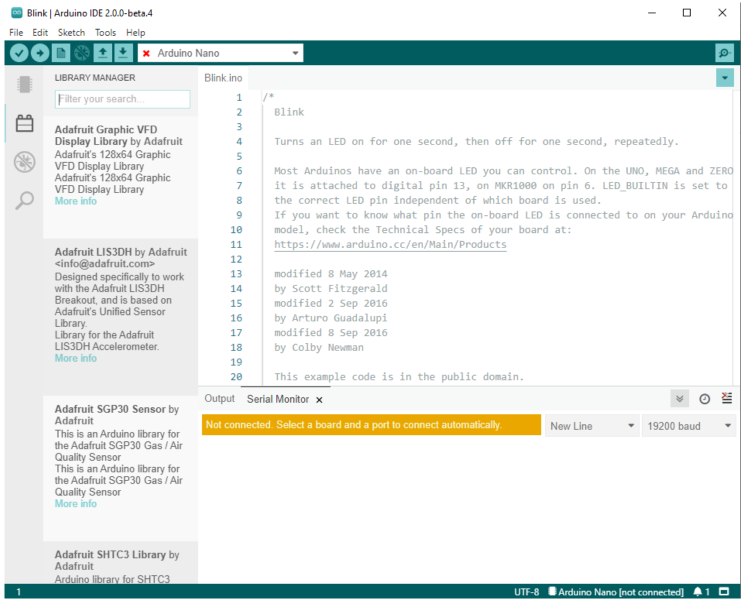
New Line | 19200 baud

1 UTF-8 Arduino Nano [not connected] 1

Arduino IDE

The code has to be written in a computer language, similar to C/C++

The Arduino programming language is optimized to be used to control a microcontroller in real time, this means that the time each instruction takes to be executed is precisely known.



Adafruit Graphic VFD Display Library by Adafruit
Adafruit's 128x64 Graphic VFD Display Library
Adafruit's 128x64 Graphic VFD Display Library
More info

Adafruit LIS3DH by Adafruit <info@adafruit.com>
Designed specifically to work with the Adafruit LIS3DH Breakout, and is based on Adafruit's Unified Sensor Library.
Library for the Adafruit LIS3DH Accelerometer.
More info

Adafruit SGP30 Sensor by Adafruit
This is an Arduino library for the Adafruit SGP30 Gas / Air Quality Sensor
This is an Arduino library for the Adafruit SGP30 Gas / Air Quality Sensor
More info

Adafruit SHTC3 Library by Adafruit
Arduino library for SHTC3

```
/*
  Blink

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

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  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.
```

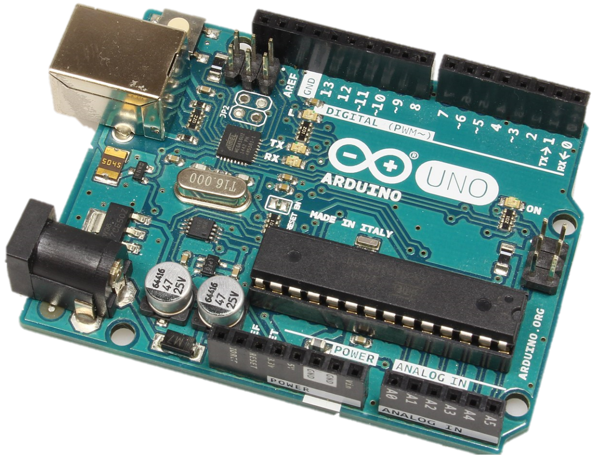
Not connected. Select a board and a port to connect automatically.

New Line | 19200 baud

1 UTF-8 Arduino Nano [not connected] 1

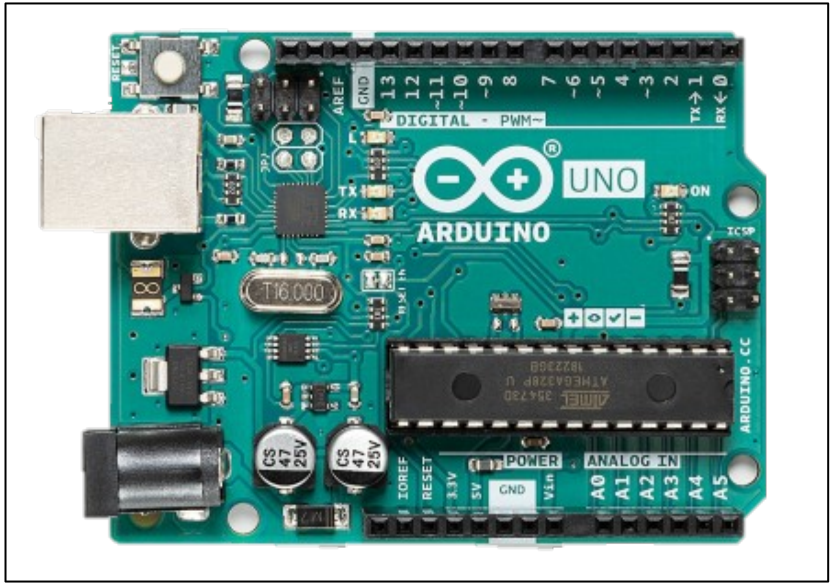
Arduino code

- Code similar to C/C++
- HUGE amount of software, documentations, examples and libraries for every purpose



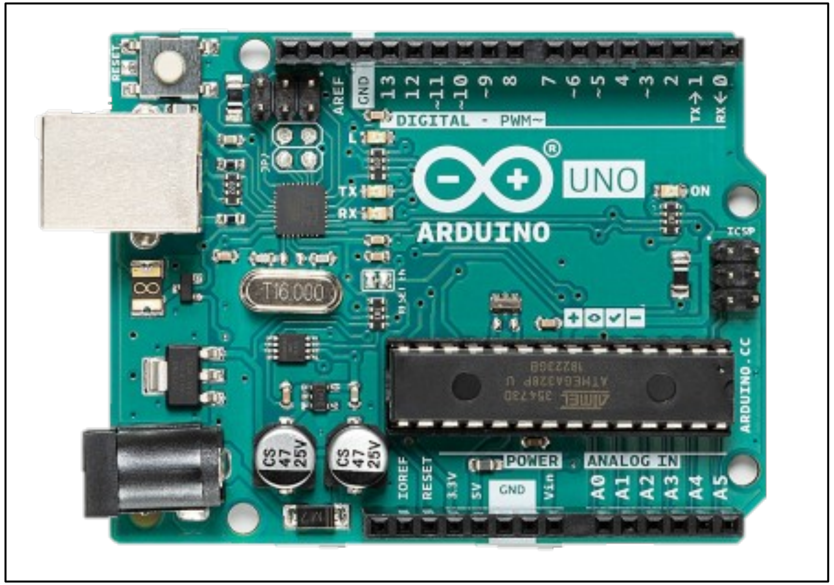
Arduino UNO

- Microcontroller: ATmega328P
- Digital Pin IO: 14
- Analog Pin I: 6
- PWM Pin: 6
- Working voltage: 5V
- Max current on Digital Pin: 20 mA
- Flash Memory: 32KB
- SRAM Memory: 2KB
- EEPROM Memory: 1KB
- Clock: 16 MHz
- Led built-in: D13
- Native USB: no
- ADC: 1x 10bit
- DAC: no
- Time/Counter: 2x 8bit, 1x 16bit



Arduino UNO

- Microcontroller: ATmega328P
- Digital Pin IO: 14
- Analog Pin I: 6
- PWM Pin: 6
- Working voltage: 5V
- Max current on Digital Pin: 20 mA
- Flash Memory: 32KB
- SRAM Memory: 2KB
- EEPROM Memory: 1KB
- Clock: 16 MHz
- Led built-in: D13
- Native USB: no
- ADC: 1x 10bit
- DAC: no
- Time/Counter: 2x 8bit, 1x 16bit



A lot of information!

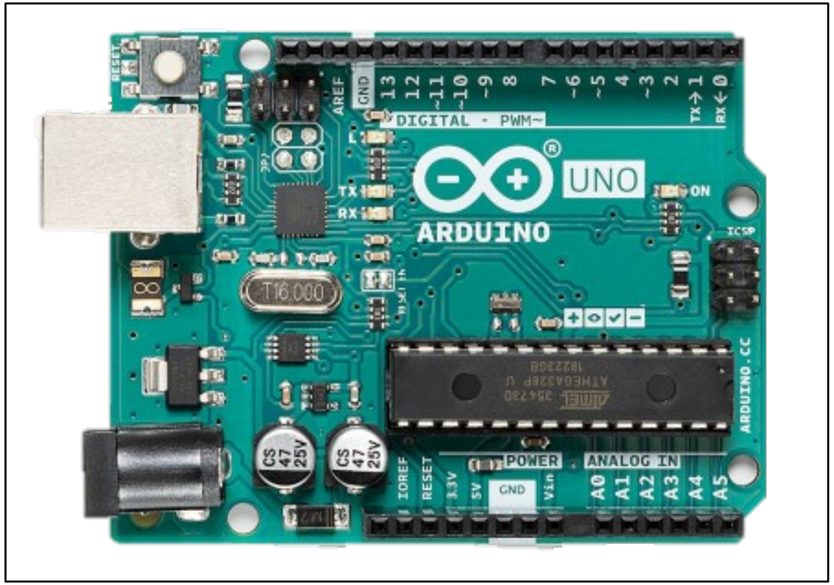
Do we need to know all these data?

It depends on our project!

Arduino UNO

- Microcontroller: ATmega328P
- Digital Pin IO: 14
- Analog Pin I: 6
- PWM Pin: 6
- Working voltage: 5V
- Max current on Digital Pin: 20 mA
- Flash Memory: 32KB
- SRAM Memory: 2KB
- EEPROM Memory: 1KB
- Clock: 16 MHz
- Led built-in: D13
- Native USB: no
- ADC: 1x 10bit
- DAC: no
- Time/Counter: 2x 8bit, 1x 16bit

Most common criteria



A lot of information!

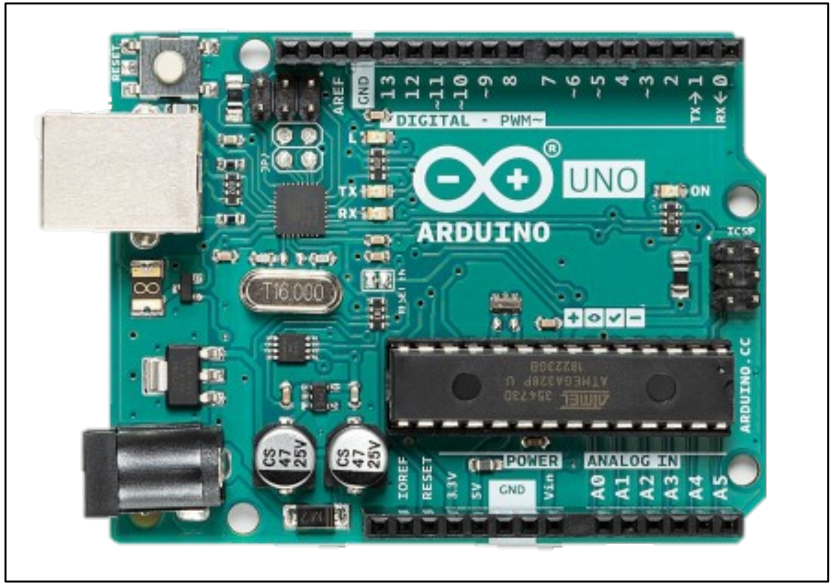
Do we need to know all these data?

It depends on our project!

Arduino UNO

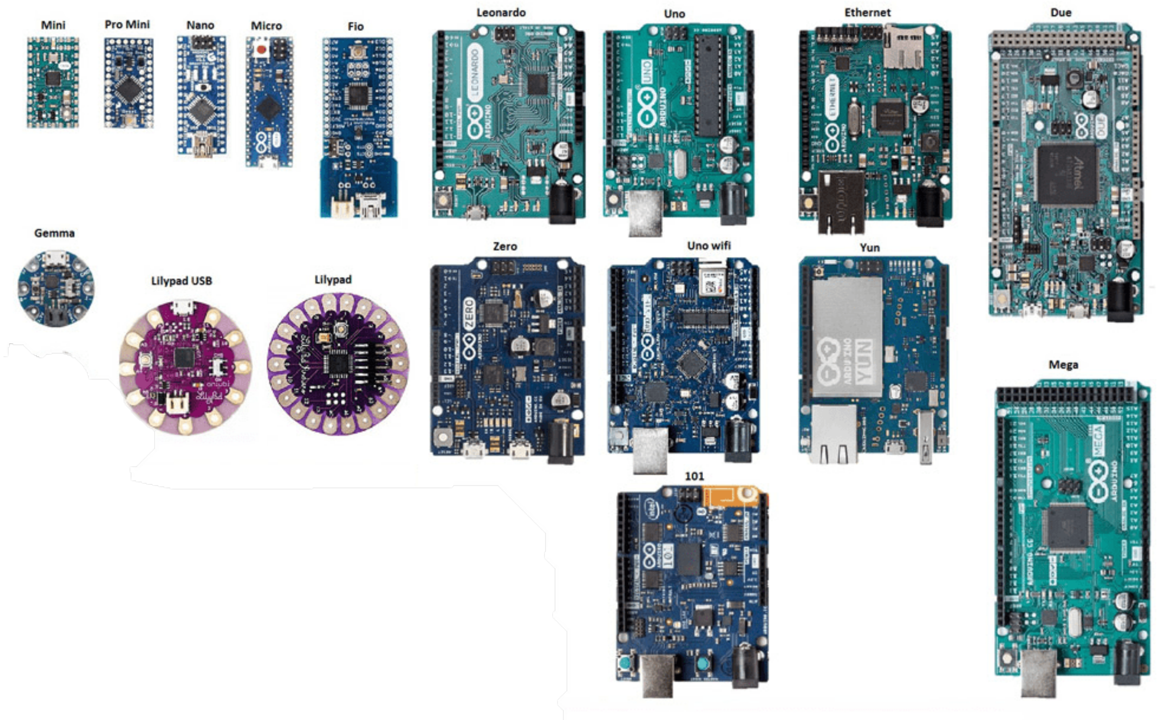
- Microcontroller: ATmega328P
- Digital Pin IO: 14
- Analog Pin I: 6
- PWM Pin: 6
- Working voltage: 5V
- Max current on Digital Pin: 20 mA
- Flash Memory: 32KB
- SRAM Memory: 2KB
- EEPROM Memory: 1KB
- Clock: 16 MHz
- Led built-in: D13
- Native USB: no
- ADC: 1x 10bit
- DAC: no
- Time/Counter: 2x 8bit, 1x 16bit

Most common criteria



What if these IO pins are not enough?

Arduino big family

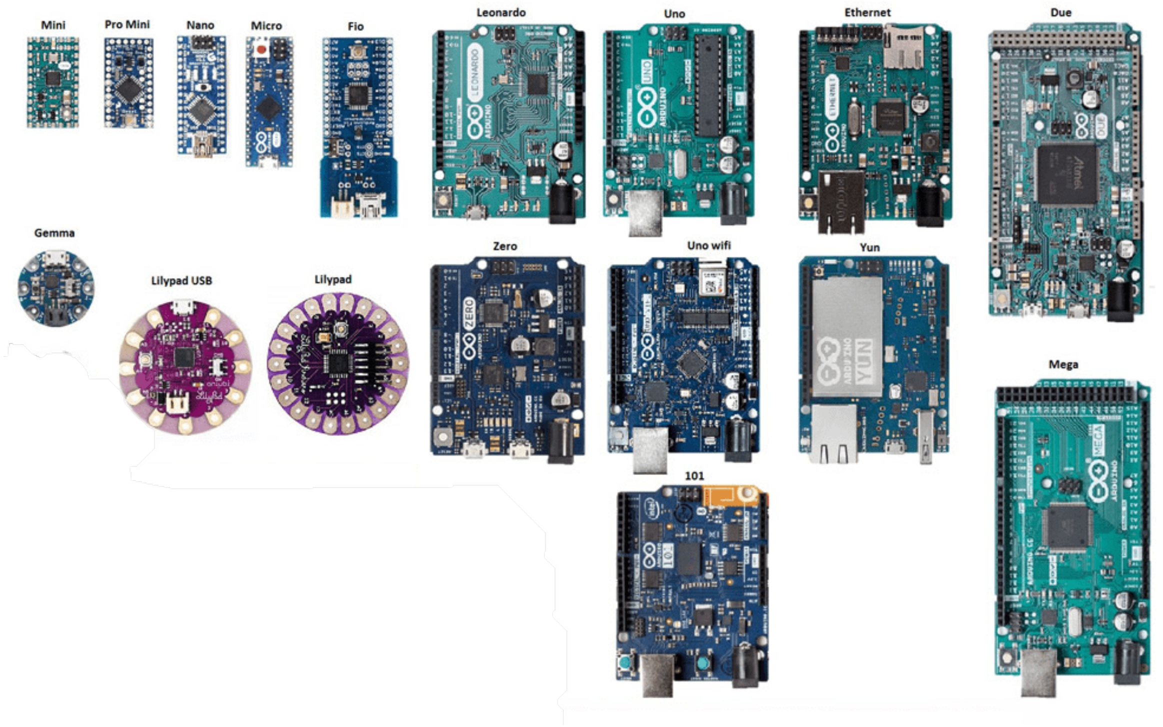


DiecimilaDue
Duemilanove
Uno
Leonardo
Mega
Fio
Nano
LilyPad
Yun
101
M0 Pro
Zero
Nano 33 IoT
Nano 33 BLE
Nano Every
Micro
MKR1000
MKR Zero
…and many others!

The list is very long

Which are your project requirements?

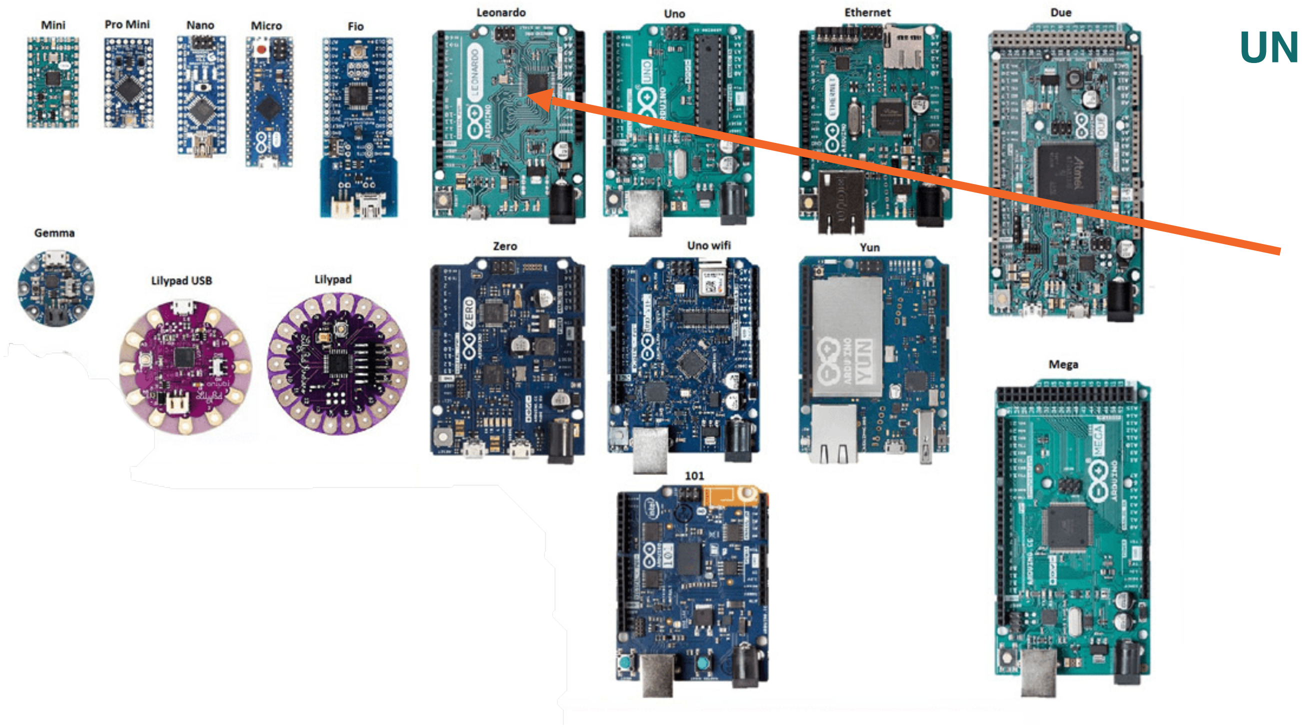
Arduino big family



An example:
UNO has 13 digital pins

More pins?

Arduino big family

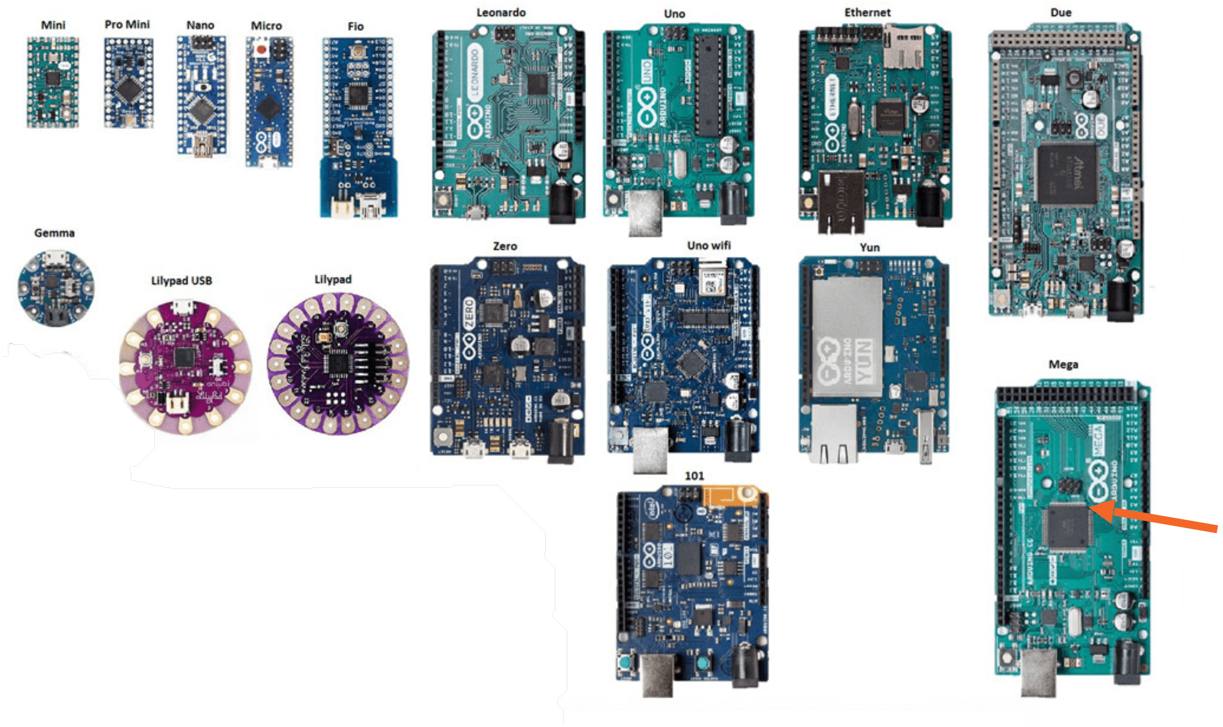


An example:
UNO has 13 digital pins

More pins?

Leonardo has
20 digital IO pins

Arduino big family



An example:
UNO has 13 digital pins

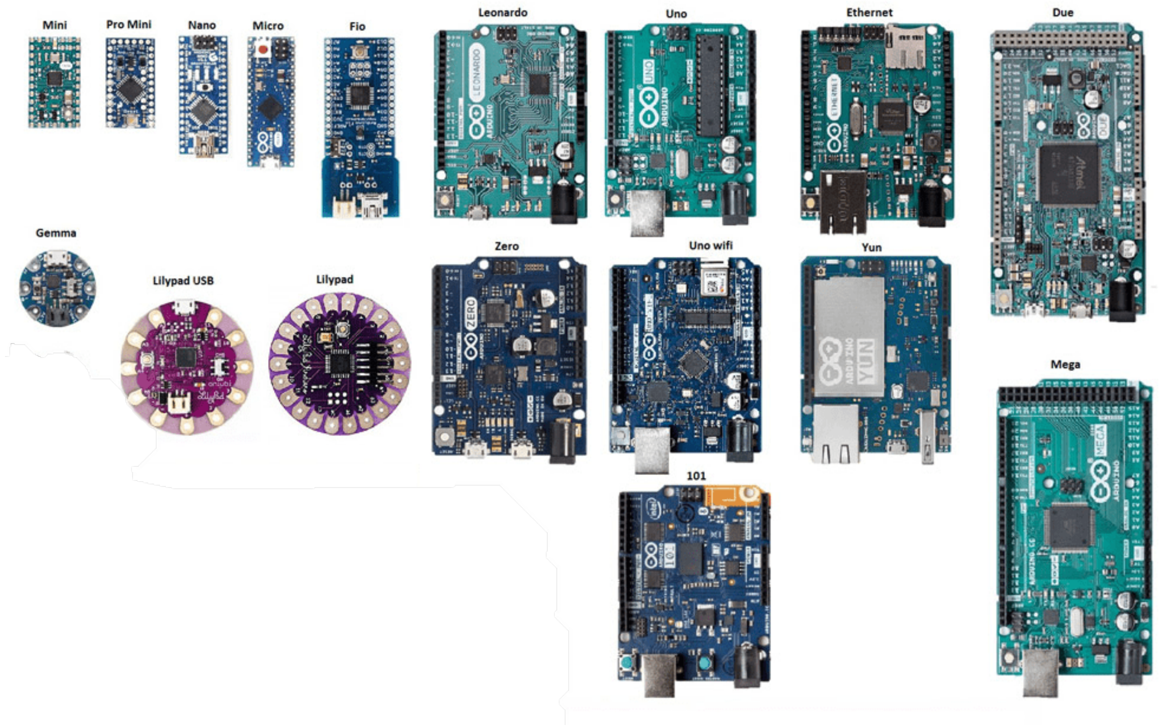
More pins?

Leonardo has 20 digital IO pins

More Pins?

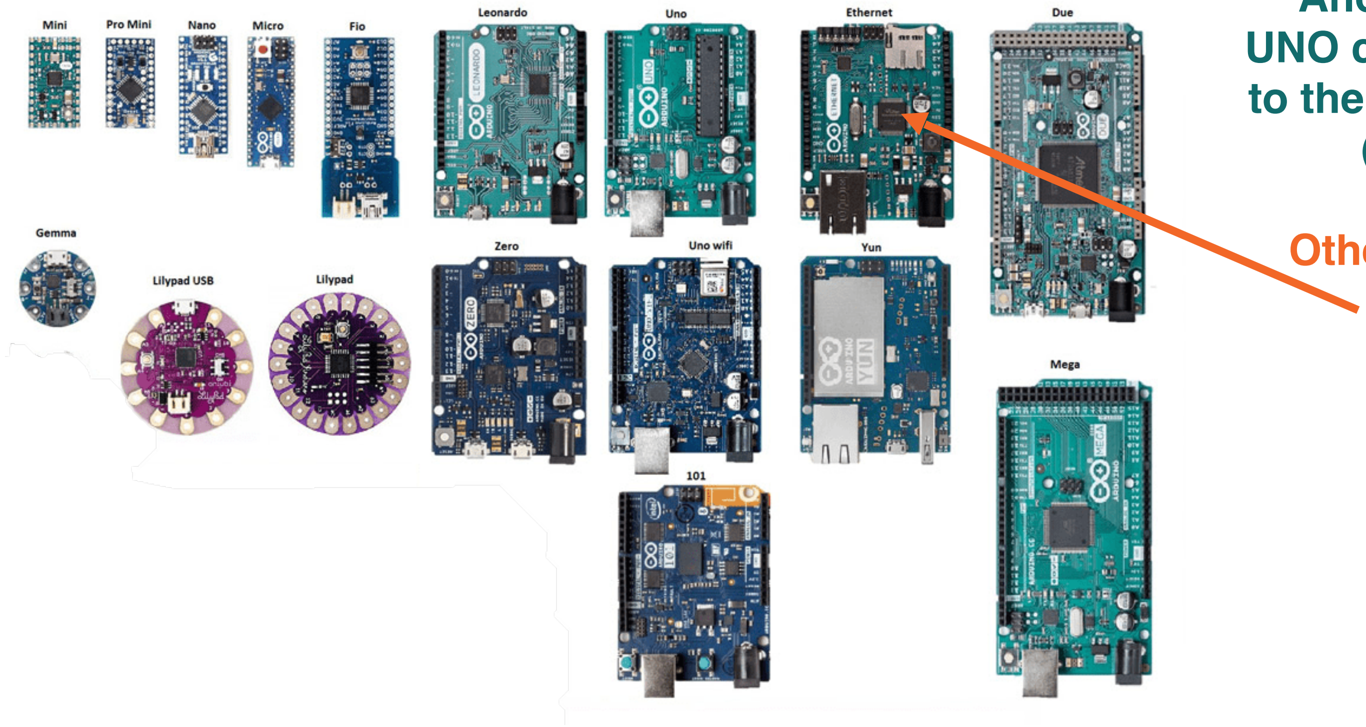
Mega has 54 Digital IO pins!

Arduino big family



**Another example:
UNO can be connected to the PC only via USB (serial port)**

Arduino big family

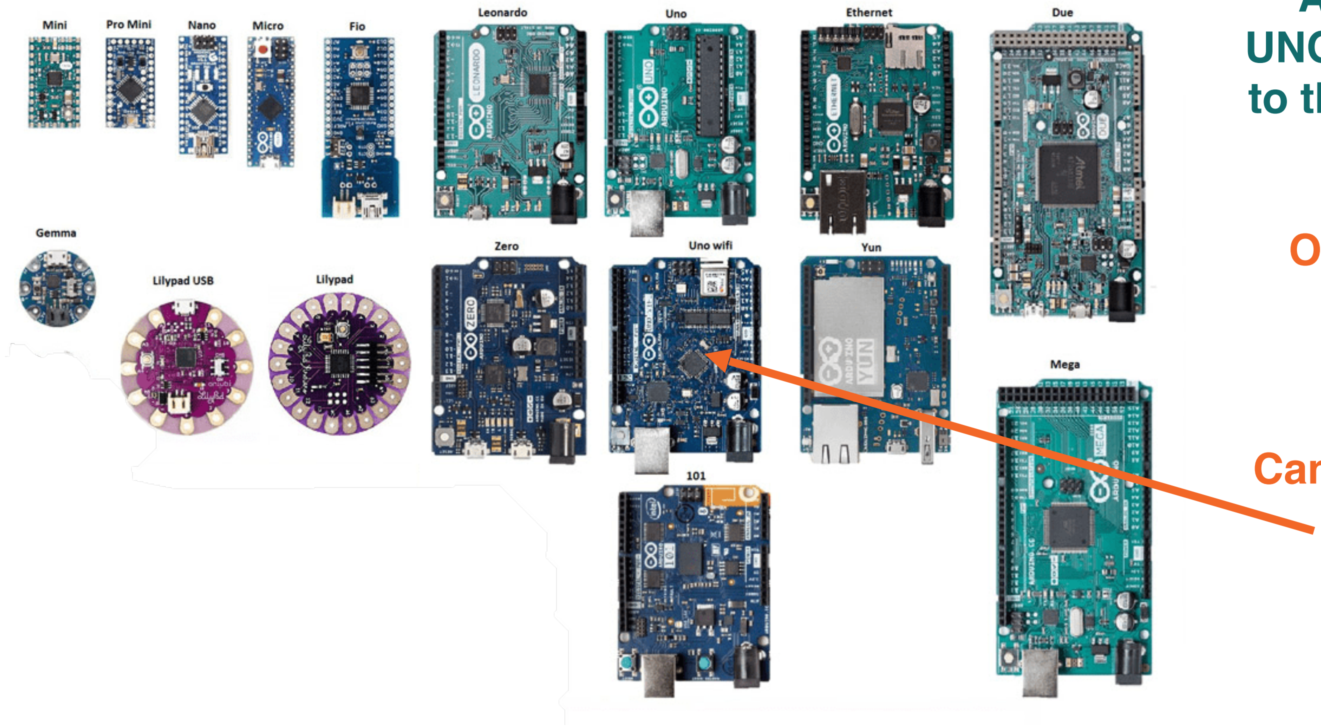


Another example: UNO can be connected to the PC only via USB (serial port)

Other connection?

Ethernet!

Arduino big family



Another example: UNO can be connected to the PC only via USB (serial port)

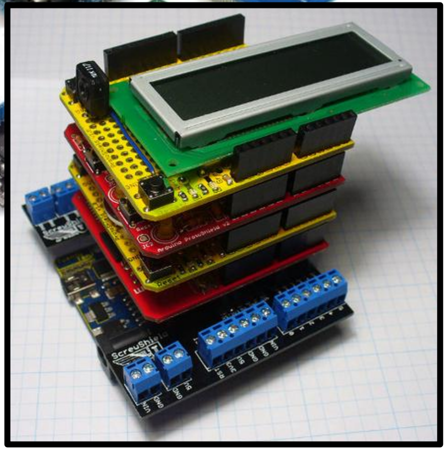
Other connection?

Ethernet!

Cannot run the cable?

Uno WIFI!

Do you need more features?



Use a shield! Shields are also called daughter boards.

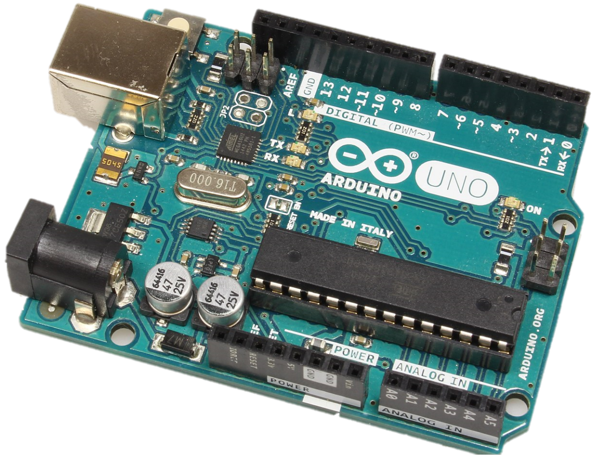
Stackable electronic boards that can be connected to the Arduino (standard header layout).

Different versions for different Arduino boards and for different requirements:

- Displays
- Controllers
- Motor drivers
- Sensors
- Connections

These are basic Arduinos…

The Arduino boards that I showed you are the basics…
They are our allies in our small project.
What if I need something more powerful?

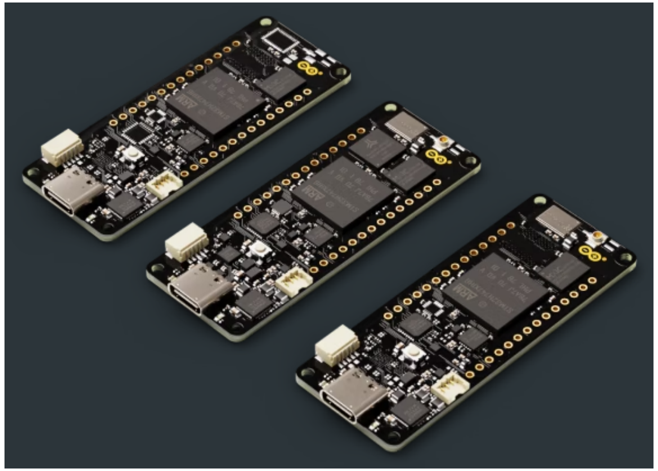


Arduino MKR

- Family of small boards with more advanced CPU than the basic Arduinos
- Designed for IOT
- Different sensors or receivers (GPS)
- They have some type of wireless connection for communications (WiFi, Lora, etc)

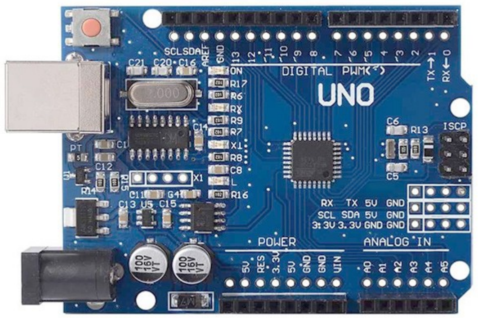
Arduino Portenta

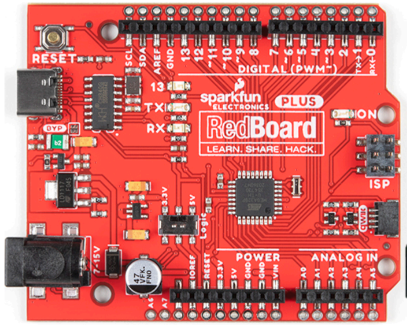
- Powerful SOC (system on chip)
- It is like a tiny computer (it also runs Linux)
- You can run AI stuff and other programs that require more resources
- Developed for fast industrial R&D

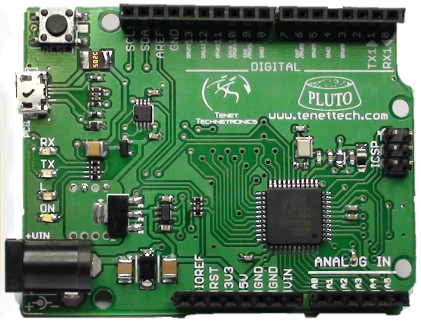


Arduino clones

Arduino hardware is mostly opensource/openhardware,
It can be made by everyone!







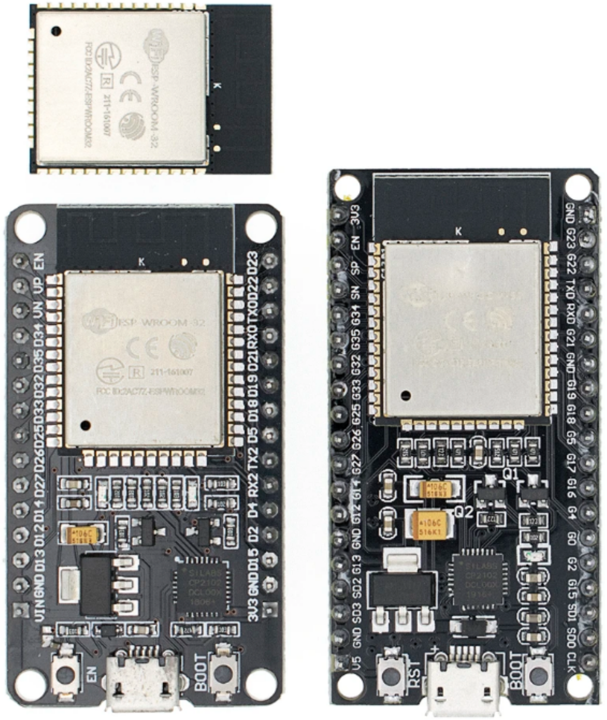
Not clone, but alternatives?

Nowadays there are a lot of competitors with different spec.

The most common are the **ESP32.**

Also worth mentioning are the fairly recent **Pyboards**, i.e. boards supporting **Python** language (loved by scientists)

Raspberry Pi anyone? =)

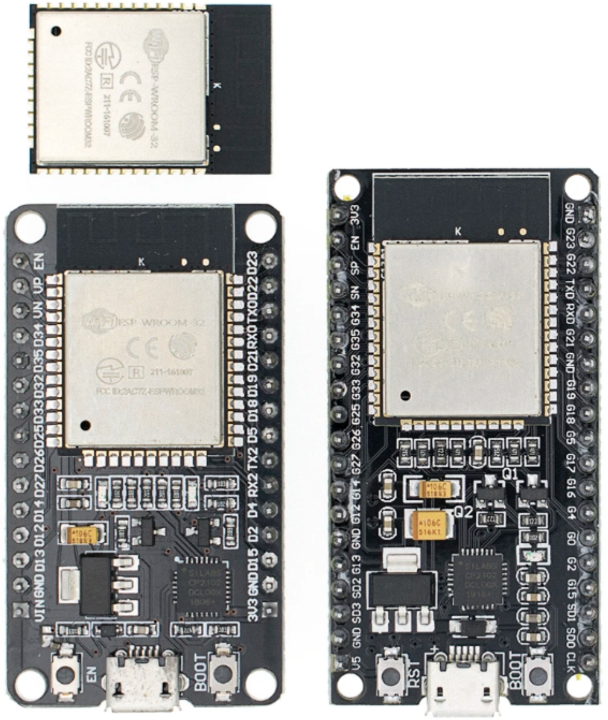


ESP32

It is a SOC microcontroller very common, with a very large community and very cheap (10$).

It has:

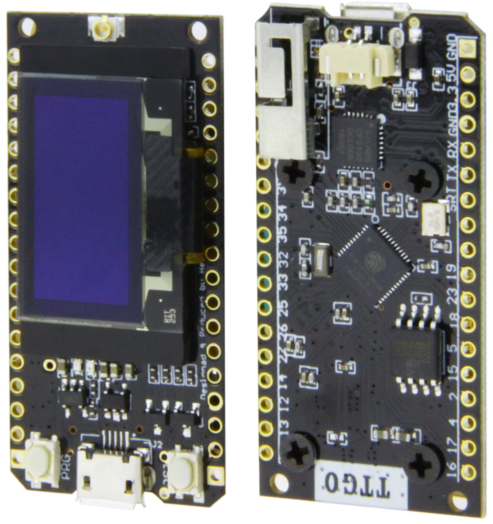
- X32 core (some version dual)
- WiFi
- Bluetooth
- Up to 34 GPIO

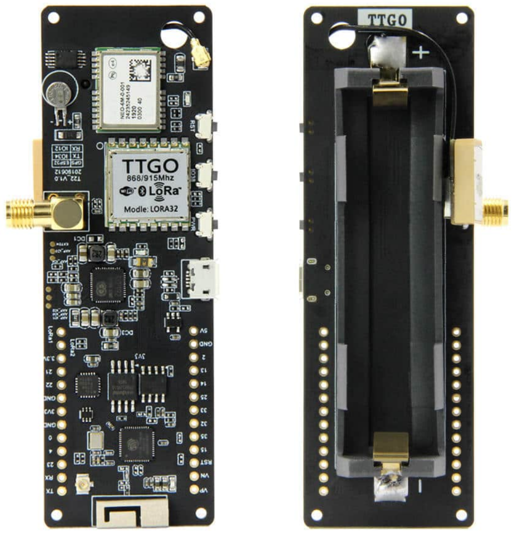


To be mentioned: TTGO

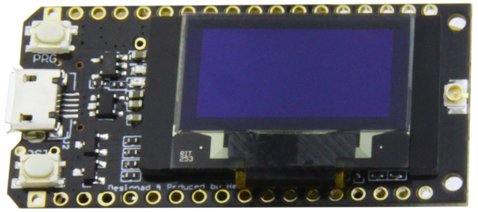
Based on ESP32, it is a series of board which comprehend different versions with different setup:

- Built in OLED screen
- Power circuit
- LoRa chips
- GPS receiver
- SD card reader



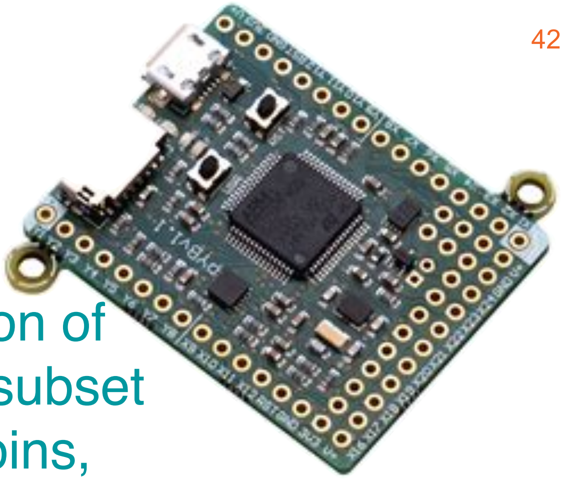


Examples: advanced projects with ESP32



- LoRa satellite receiver (tinygs.com)
- Weather balloons receiver (mysondy.altervista.org)
- LoRa long range network (meshtastic.org)

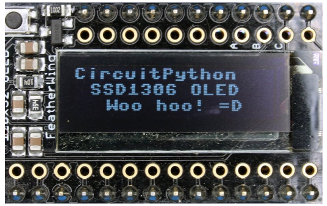
Pyboards

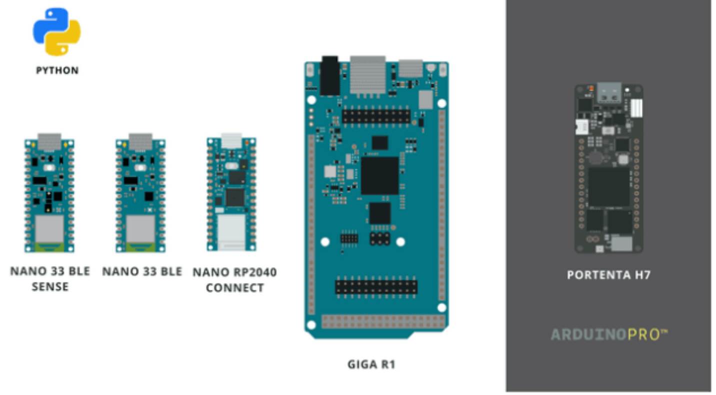


Pyboards are a family of small boards that can be programmed with **MicroPython**, a “simplified” version of Python language (compatible with Python 3, with a subset of libraries and some added features to control I/O pins, etc). The code is uploaded to the board via a simple file copy/paste to a USD drive. Then, it is interpreted by the firmware running on the microprocessor.

https://micropython.org

There is also the similar CircuitPython by Adafruit.

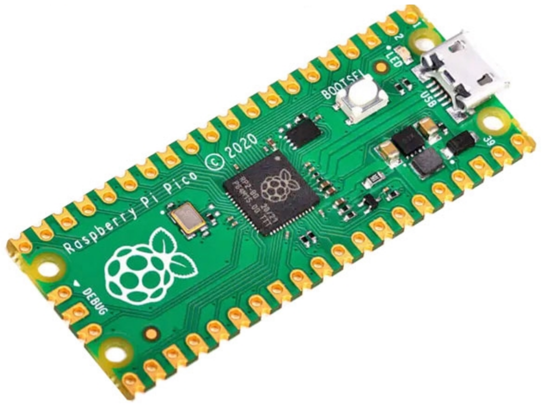
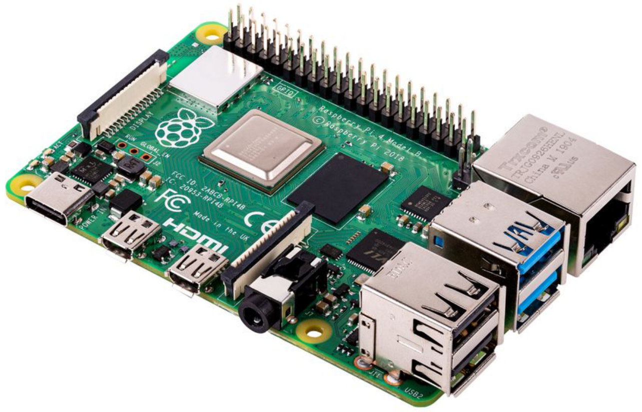




I heard of Raspberry PI…

It is another known brand, they produce two different types of board:

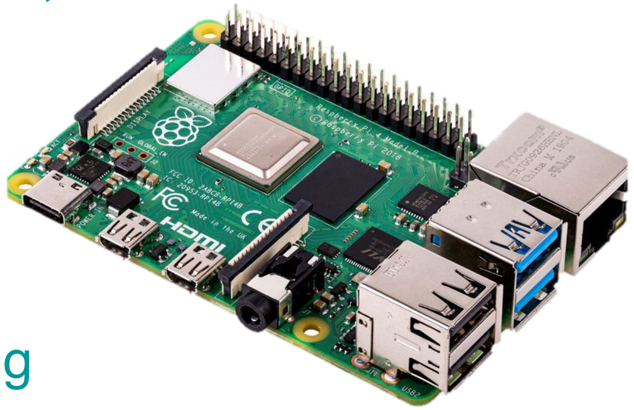
- Raspberry Pi Pico
- Raspberry Pi 3B-4-5-Zero

Raspberry PI 3B-4-Zero

These are like real computer, they run Linux and are useful for an incredible number of applications, like:

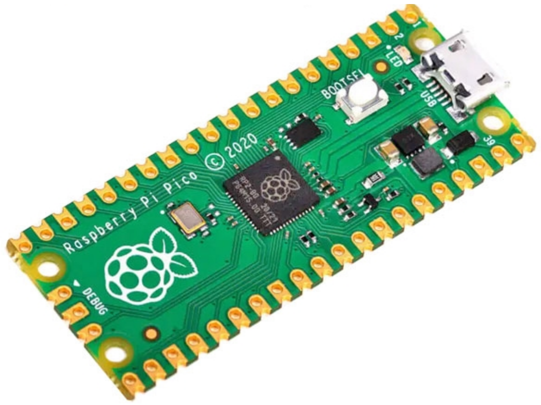
- Low cost desktop computer
- Video/multimedia player
- Small server
- ….



Example: use them to collect data for tracking airplanes =)

Raspberry Pi Pico

It is like an Arduino,
in some way similar to a MKR

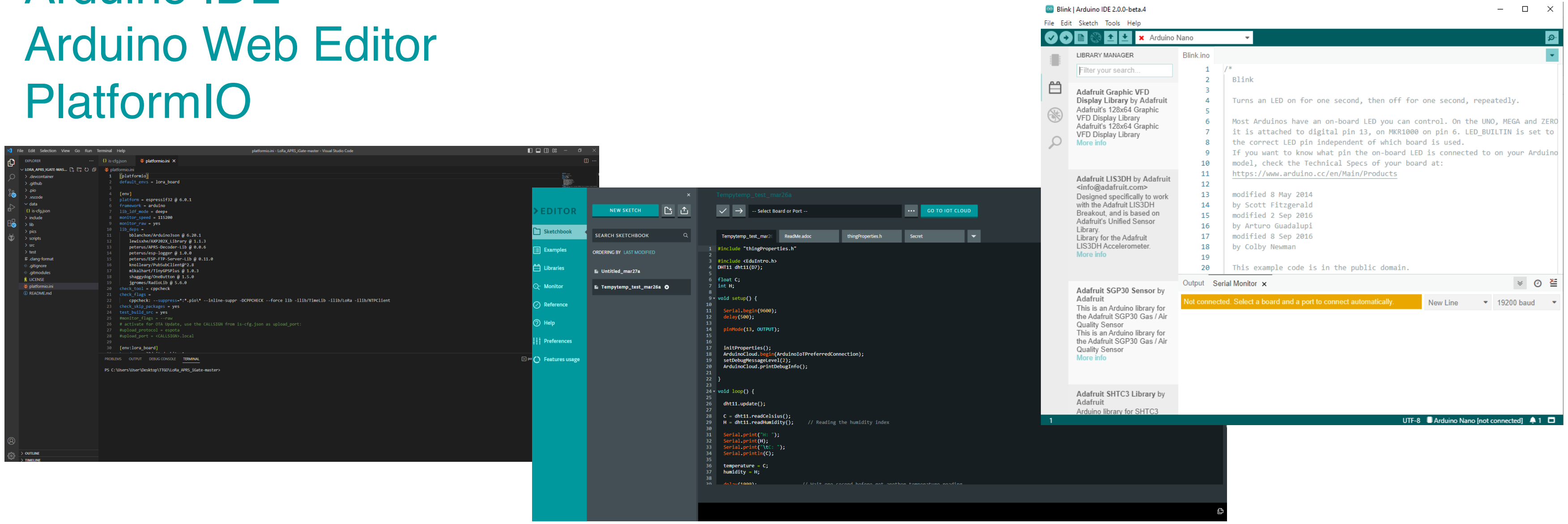


(… as you saw, there are so many, and this one is still on my list for testing!)

How to program an Arduino

The three most used platforms for programming:

- Arduino IDE
- Arduino Web Editor
- PlatformIO



platformio.ini - LoRa_APRS_iGate-master - Visual Studio Code

```
[platformio]
default_envs = lora_board

[env]
platform = espressif32 @ 6.0.1
framework = arduino
lib_ldf_mode = deep+
monitor_speed = 115200
monitor_raw = yes
lib_deps =
    bblanchon/ArduinoJson @ 6.20.1
    lewisxhe/AXP202X_Library @ 1.1.3
    peterus/APRS-Decoder-Lib @ 0.0.6
    peterus/esp-logger @ 1.0.0
    peterus/ESP-FTP-Server-Lib @ 0.11.0
    knolleary/PubSubClient@^2.8
    mikalhart/TinyGPSPlus @ 1.0.3
    shaggydog/OneButton @ 1.5.0
    jgromes/RadioLib @ 5.6.0
check_tool = cppcheck
check_flags =
    cppcheck: --suppress=*:*.pio\* --inline-suppr -DCPPCHECK --force lib -ilib/TimeLib -ilib/LoRa -ilib/NTPClient
check_skip_packages = yes
test_build_src = yes
#monitor_flags = --raw
# activate for OTA Update, use the CALLSIGN from is-cfg.json as upload_port:
#upload_protocol = espota
#upload_port = <CALLSIGN>.local

[env:lora_board]
```

```
PS C:\Users\User\Desktop\TTGO\LoRa_APRS_iGate-master>
```

Tempytemp_test_mar26a

```
#include "thingProperties.h"

#include <EduIntro.h>
DHT11 dht11(D7);

float C;
int H;

void setup() {

  Serial.begin(9600);
  delay(500);

  pinMode(13, OUTPUT);


  initProperties();
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();

}

void loop() {

  dht11.update();

  C = dht11.readCelsius();
  H = dht11.readHumidity();     // Reading the humidity index

  Serial.print("H: ");
  Serial.print(H);
  Serial.print("\tC: ");
  Serial.println(C);

  temperature = C;
  humidity = H;
```

Blink | Arduino IDE 2.0.0-beta.4

LIBRARY MANAGER

Adafruit Graphic VFD Display Library by Adafruit
Adafruit's 128x64 Graphic VFD Display Library
Adafruit's 128x64 Graphic VFD Display Library
More info

Adafruit LIS3DH by Adafruit <info@adafruit.com>
Designed specifically to work with the Adafruit LIS3DH Breakout, and is based on Adafruit's Unified Sensor Library.
Library for the Adafruit LIS3DH Accelerometer.
More info

Adafruit SGP30 Sensor by Adafruit
This is an Arduino library for the Adafruit SGP30 Gas / Air Quality Sensor
This is an Arduino library for the Adafruit SGP30 Gas / Air Quality Sensor
More info

Adafruit SHTC3 Library by Adafruit
Arduino library for SHTC3

Blink.ino

```
/*
  Blink

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

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  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.
```

Not connected. Select a board and a port to connect automatically.

New Line · 19200 baud

Now we will move to the hands-on part of today's "Arduino 101" event:
First we will install the Arduino IDE and then Gaya will guide us trough a few hands-on exercises with an Arduino Uno and some I/O electronic components, like LEDs and sensors!