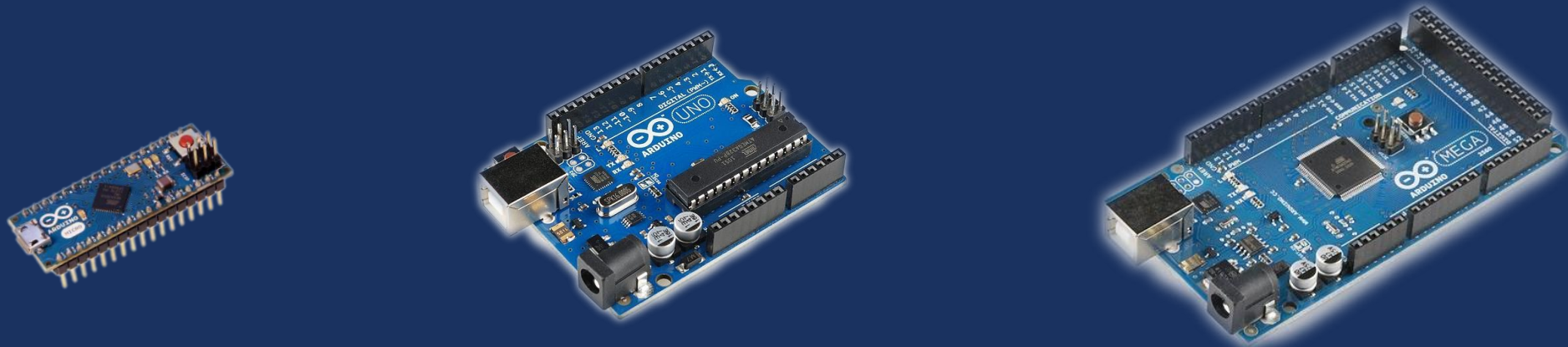# INTRO TO MICROCONTROLLERS WITH ARDUINO

JULIEN DE LA BRUÈRE-TERREAULT

MICROCONTROLLERS AND DEVELOPMENT BOARDS

Microcontrollers are integrated circuits (IC) that includes a central processing unit (CPU), memory and input / output peripherals (GPIO) on a single chip. Most microcontrollers also include Analog-Digital Converters (ADCs), timers and interrupts.
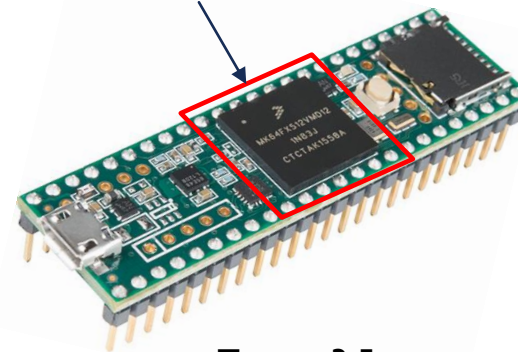
Development boards are printed circuit boards combining, in a practical format, a microcontroller with features such as :

- GPIO pins or connectors

- Voltage regulator

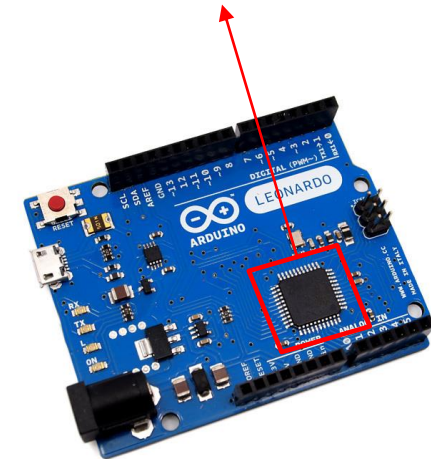- Built-in LEDs

- USB interface

- Various board specific features

ATmega 32U4
microcontroller
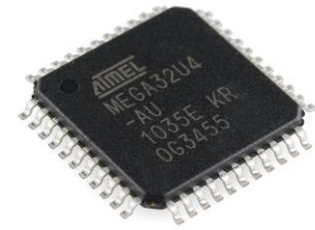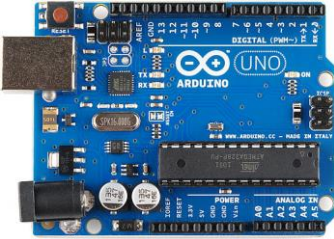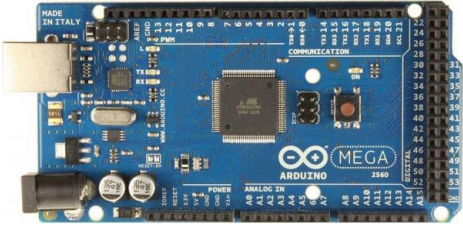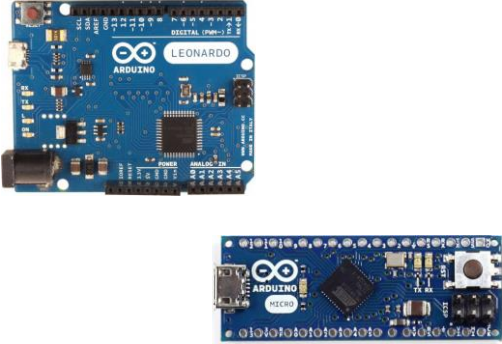
ARM Cortex M4
microcontroller
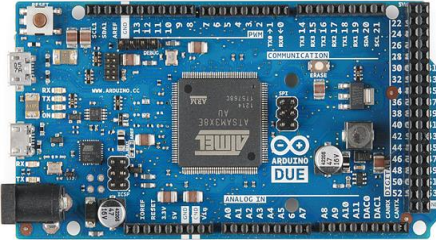
Teensy 3.5
development board

Arduino Leonardo
development board

POPULAR DEVELOPMENT BOARDS

8 bits - 5V logic level	**Arduino Uno R3** ATmega 328P	**Arduino Mega** ATmega 2560	**Arduino Leonardo/Micro** ATmega 32U4	**Teensy 2.0** ATmega 32U4
32 bits - 3.3V logic level	**Arduino Due** ARM Cortex M3	**Teensy 3.x** ARM Cortex M4	**NodeMCU** ESP2866 Wi-Fi chip / Microcontroller	**Adafruit ESP32** ESP32 Wi-Fi / Bluetooth / Microcontroller

GETTING STARTED WITH ARDUINO

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software. (Wikipedia)

Arduino resources on arduino.cc :

- Getting started guide

- Arduino introduction

- Arduino tutorials

- How it works: Foundations

- Language reference

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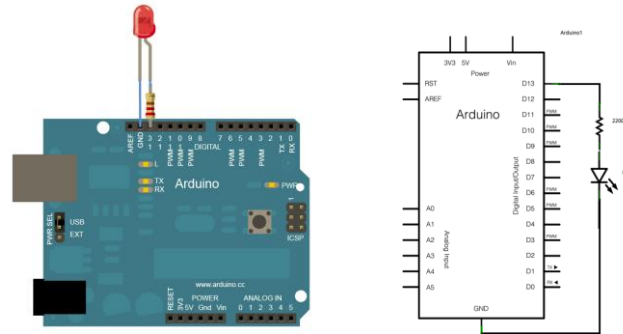
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// 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
}
```

RECOMMENDED MATERIAL

Arduino board:

- **UNO**: Most popular, versatile

- **Leonardo:** More I/Os, more interrupts than Uno

- **Micro**: Same as Leonardo in a smaller format.

Uno and Leonardo share the same form factor and are compatible with a variety of third party shields.

Micro is cheap and compact, perfect for permanent setups. Requires welding of header pins or wires directly to the board.

Leonardo and Micro can emulate a keyboard or mouse when connected to a computer.

Uno, Leonardo and Micro all have a 5V logic level.

Other material:

- USB cable

- Prototyping breadboard

- Jumper wires

- LEDs

- Resistors (220 ohm, 10K ohm)

- Potentiometers (variable resistor, 10K ohm)

- Pushbuttons

- Power supply or batteries

POWER CONSIDERATIONS

Powering the Arduino (5V boards):

- USB power: provides enough power for the Arduino and low current peripherals (e.g. single LEDs).

- Vin pin / barrel plug : 7-12V input powering the on-board voltage regulator.

- Power source selection is automatic

Powering peripherals:

- 5V pin: Output of on-board voltage regulator, ~250 mA max.

- 3.3V pin: Regulated output, 50 mA max.

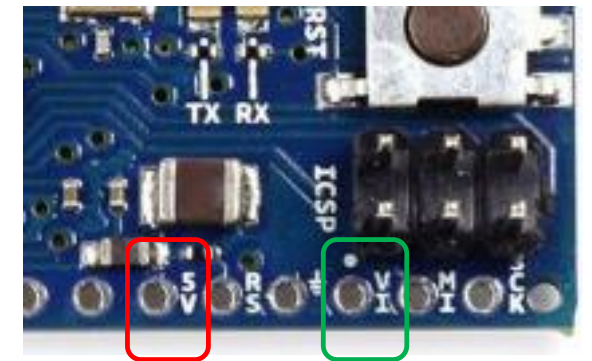- GPIO pins: 20 mA max per pin.

An external power source is recommended for peripherals drawing more than 250 mA total. Motors and servos definitively need to be powered by an external power source (power supply or batteries).

Adruino Uno / Leonardo
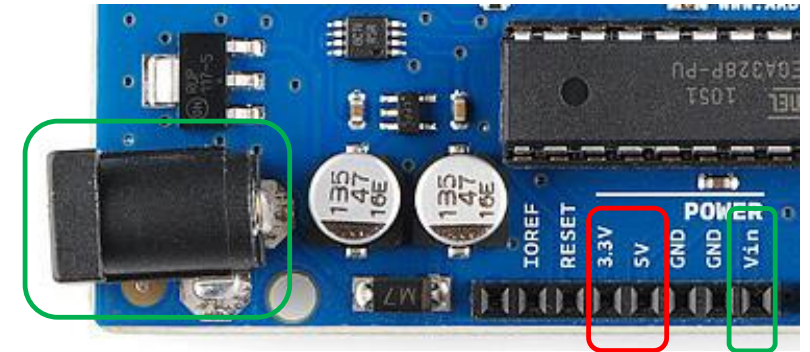
In   Out

Adruino Micro

PIN MAPPING DIAGRAMS (PINOUT)

The pinout diagrams provide information about the different functions of each pin of the microcontroller.

- Analog-Digital Converter (ADC) "Analog Pins"

- Pulse Width Modulation (PWM) enabled pins

- Serial communications (TX/RX, SDA/SCL, SPI) pins

- Interrupts and Pin Change Interrupts (INT/PCINT)

- ATmega port and bit number for direct access

The diagrams are available for each board on the Arduino products page.
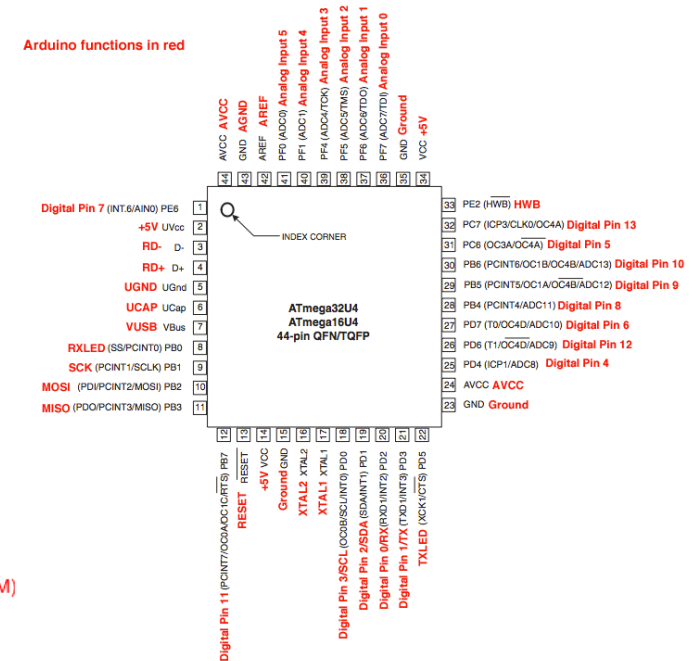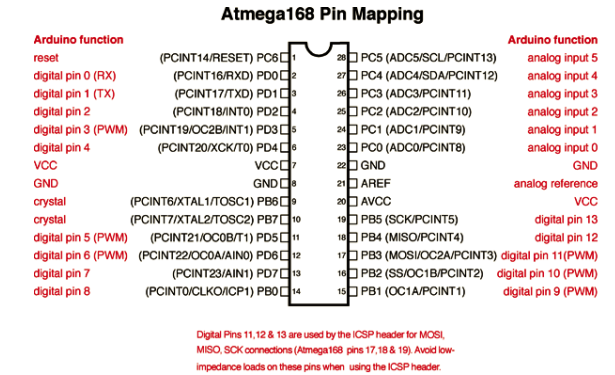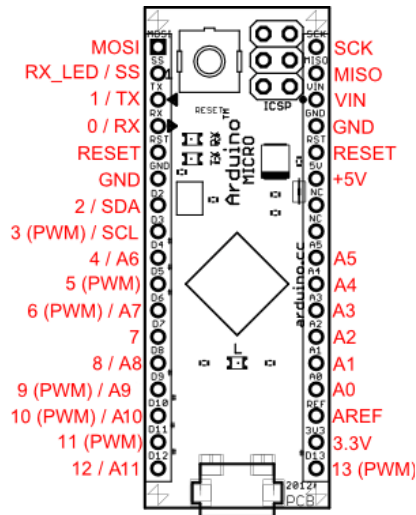
PROGRAMMING THE ARDUINO

Arduino programs are developed on a personal computer (Windows, Mac, Linux) using the Arduino IDE (Interactive Development Environment).

When ready, the program is compiled and then uploaded to the Arduino flash memory via the USB connection or a through some dedicated pins (ICSP).

The Arduino loads and runs its program as soon as it is powered.

The Arduino can run without connection to the computer but can communicate with the computer over a serial connection through the USB cable (access via the IDE's *Serial monitor*).
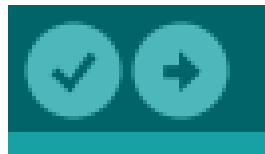
Arduino IDE

Compile and
Upload buttons
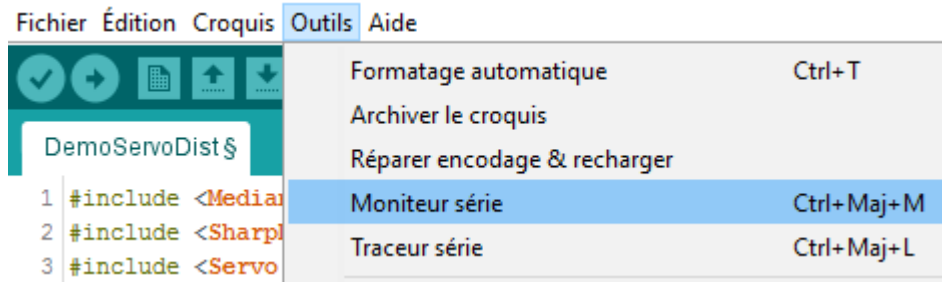
Serial monitor in the *Tools* menu

A TYPICAL ARDUINO PROGRAM (SKETCH)

The main program is stored in an Arduino "sketch" (.ino file)

The sketch can optionally include:

- Declaration of global variables

- Functions defined in the sketch file

- Functions / classes defined in a separate file within the sketch folder

- Functions / classes defined in external libraries

The sketch minimally includes two functions:

- The `setup()` function runs only once

- The `loop()` function runs continuously

BlinkWithoutDelay.ino example sketch

```
// constants won't change. Used here to set a pin number :
const int ledPin =  LED_BUILTIN;// the number of the LED pin

// Variables will change :
int ledState = LOW;             // ledState used to set the LED

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0;        // will store last time LED was updated

// constants won't change :
const long interval = 1000;           // interval at which to blink (milliseconds)

void setup() {
  // set the digital pin as output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // here is where you'd put code that needs to be running all the time.

  // check to see if it's time to blink the LED; that is, if the
  // difference between the current time and last time you blinked
  // the LED is bigger than the interval at which you want to
  // blink the LED.
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    // save the last time you blinked the LED
    previousMillis = currentMillis;

    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);
  }
}
```

WHERE TO BUY

- Buyapi.ca (CAN)   https://www.buyapi.ca/

- RobotShop (CAN)   http://www.robotshop.com/ca/en/

- Adafruit (US)   https://www.adafruit.com/

- SparkFun (US)   https://www.sparkfun.com/

- Amazon.ca

- ebay.ca