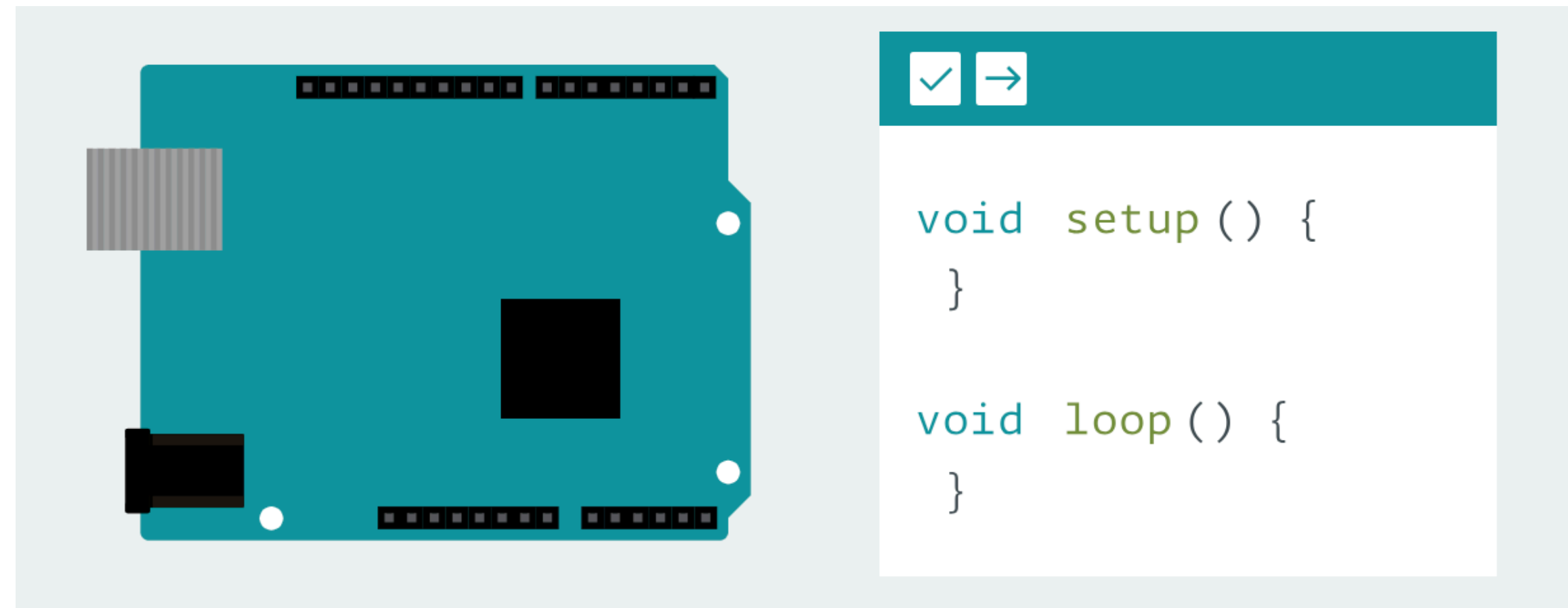


# Lecture #3

# Arduino

Spring 2024

# Foundations



1972

1982

1999

2003

# History



Arduino d'Ivrea  
(Civica raccolta stampe - Milano)



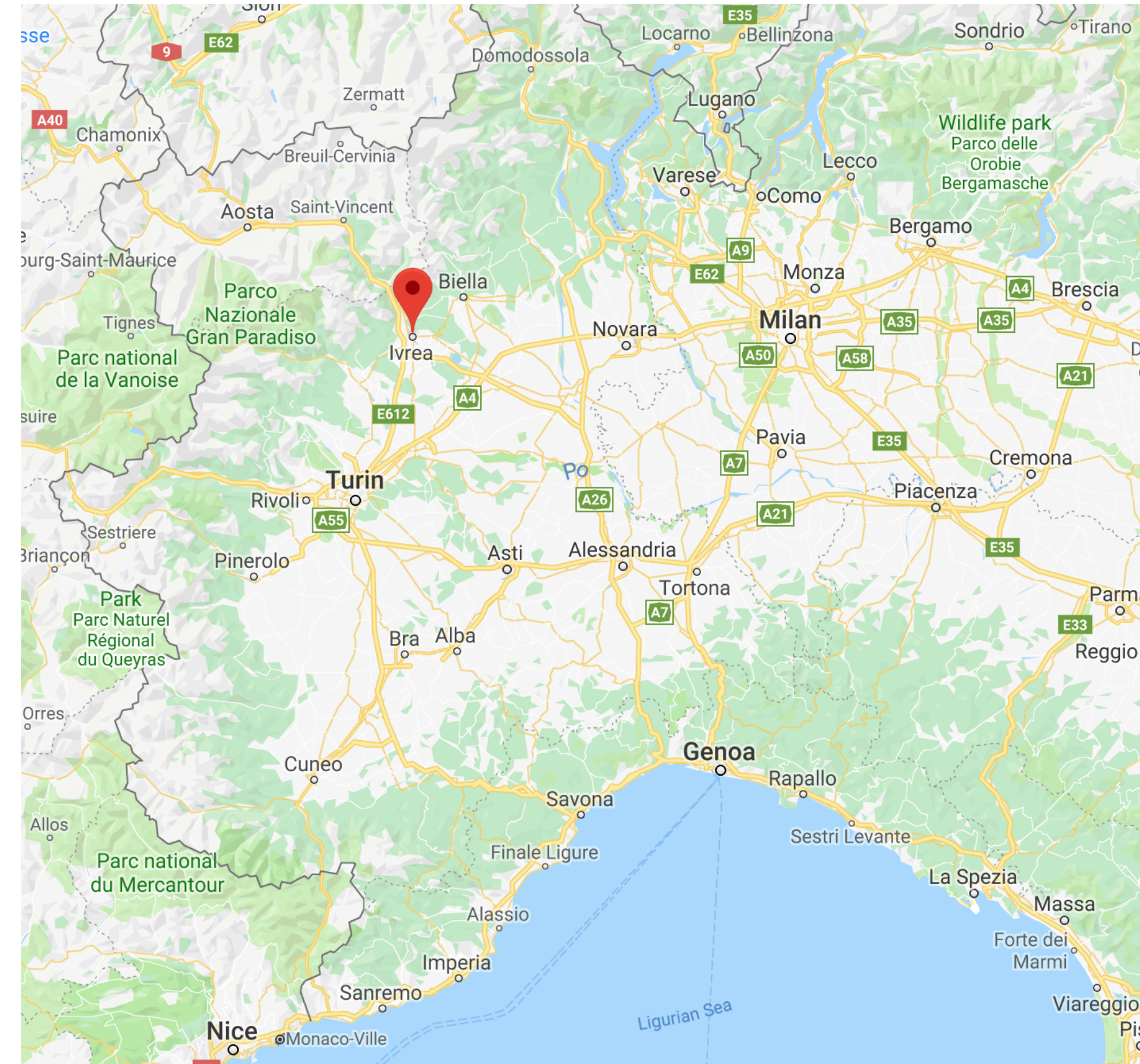
[King of Italy](#) (1002–1014)

[https://en.wikipedia.org/wiki/Arduin\\_of\\_Ivrea](https://en.wikipedia.org/wiki/Arduin_of_Ivrea)

# History



**Hernando Barragán**



**Interaction Design Institute Ivrea  
IDII**

# History



**Casey Reas**



--



**Massimo Banzi**



**2001**

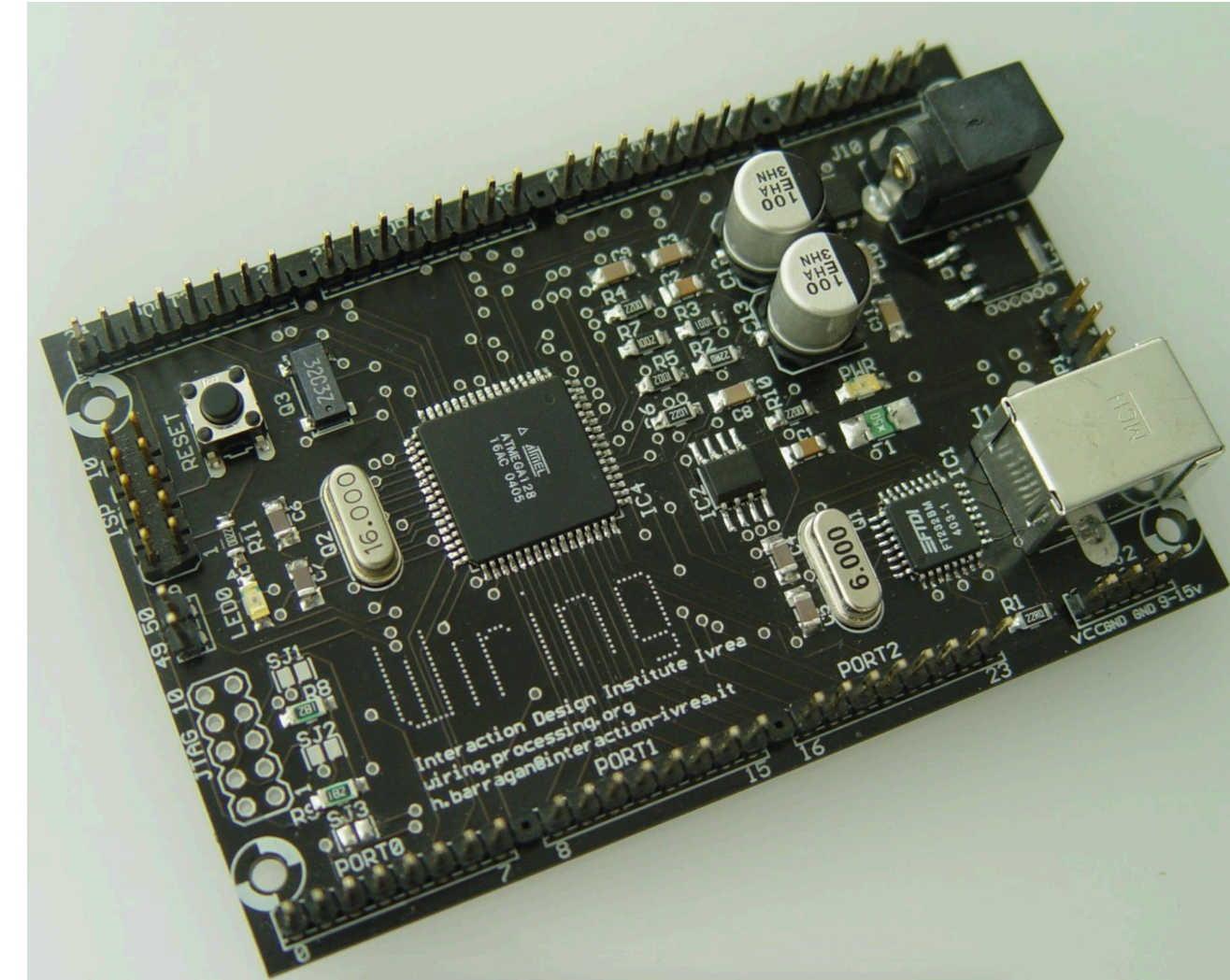
<https://processing.org/>

# History



*Wiring*

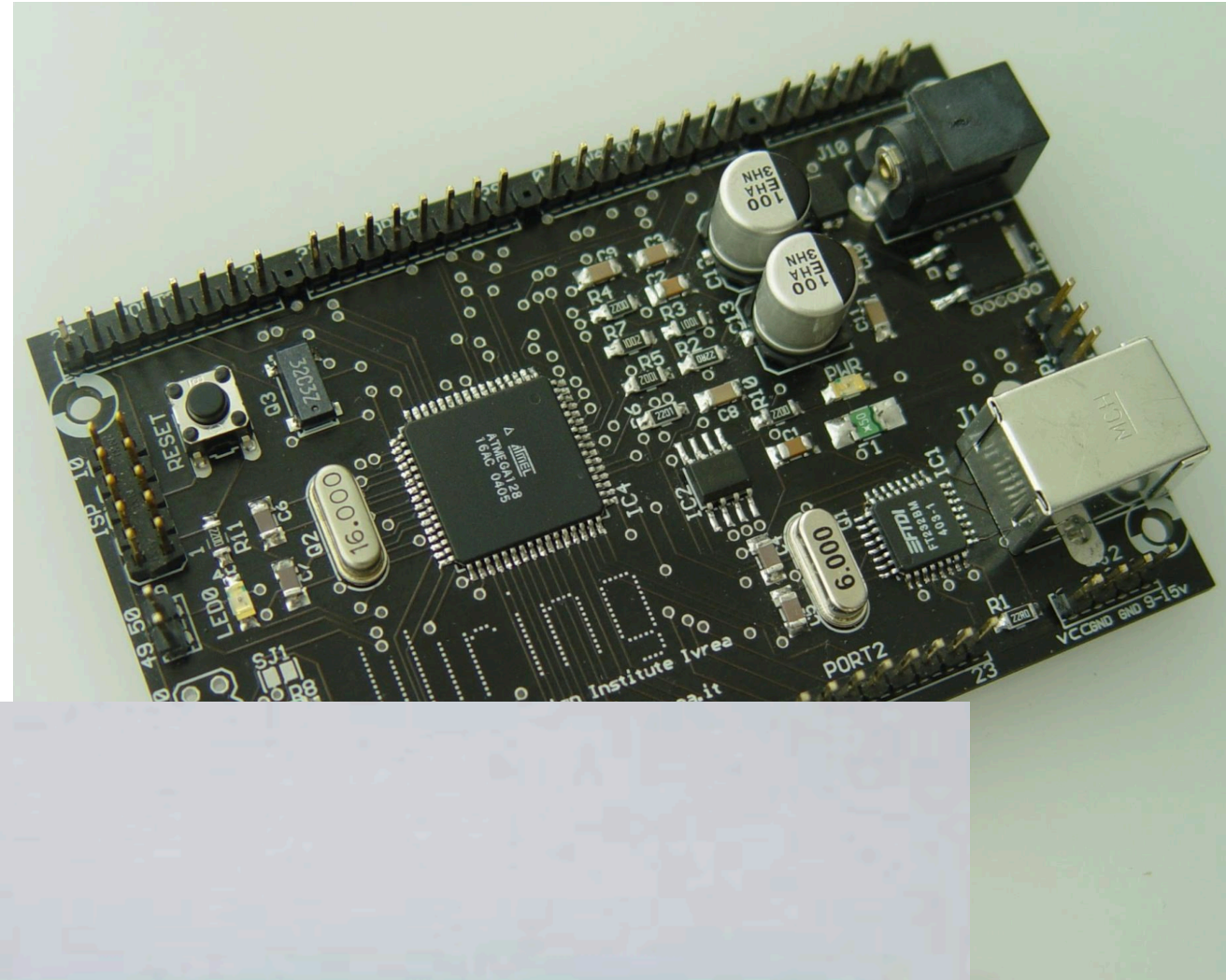
2003



```
int void setup() { // a name for the on-board LED
  ledPin = WLED;
  size(400, 400);
  stroke(255);
}
void setup() {
  background(192, 64, 0); // configure the pin for digital output
  pinMode(ledPin, OUTPUT);
}

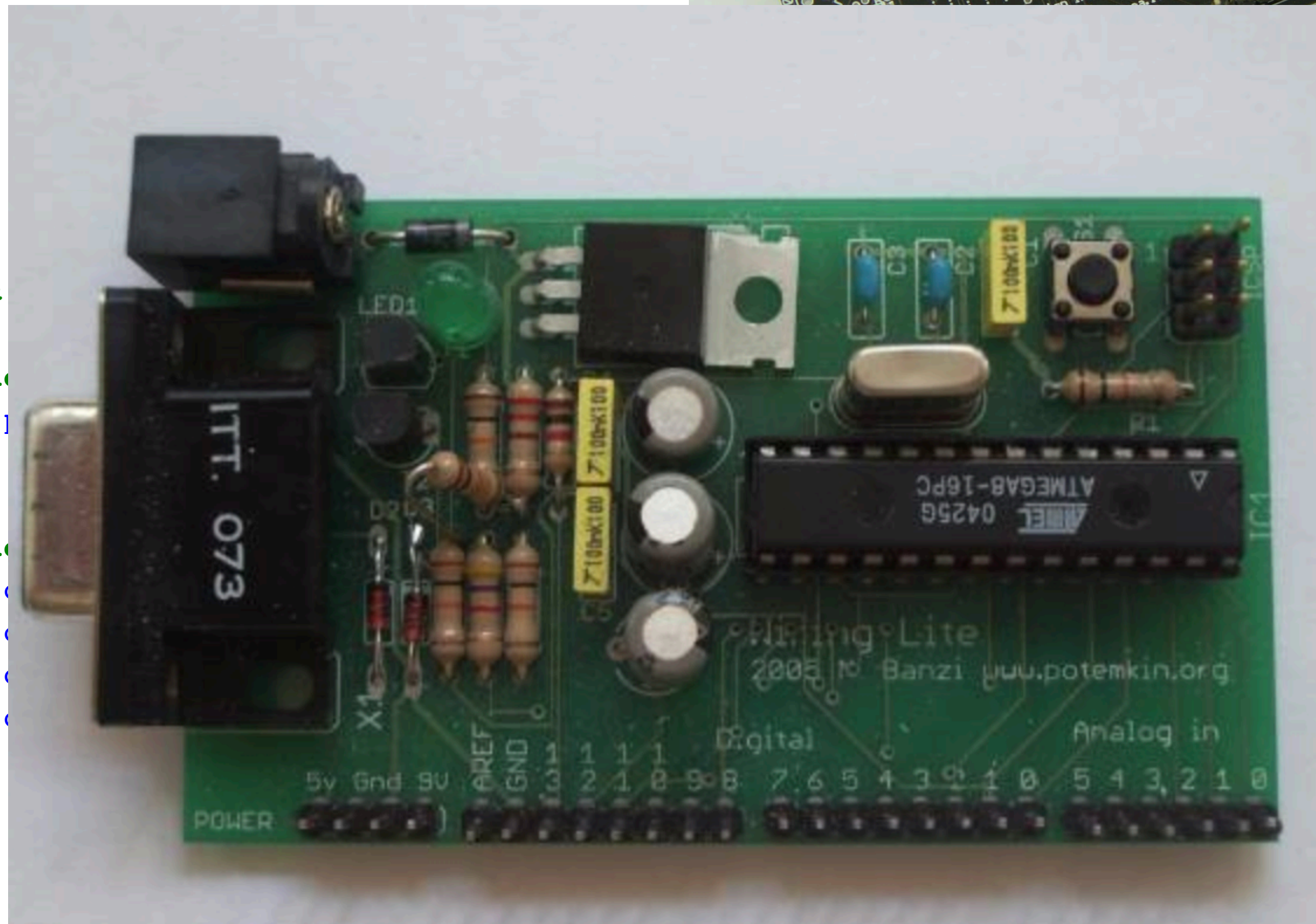
void void draw() {
  void loop() {
    line(150, 25, mouseX, mouseY); // turn on the LED
    digitalWrite(ledPin, HIGH);
    delay(1000); // wait one second (1000 milliseconds)
    digitalWrite(ledPin, LOW); // turn off the LED
    delay(1000); // wait one second
  }
}
```

# History



Wiri *Wiring*

2003



2004

Arduino

<https://arduinohistory.github.io/>

# Credits

Originally started as a research project by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis at the Interaction Design Institute of Ivrea in the early 2000s, it builds upon the Processing project, a language for learning how to code within the context of the visual arts developed by Casey Reas and Ben Fry as well as a thesis project by Hernando Barragan about the Wiring board.

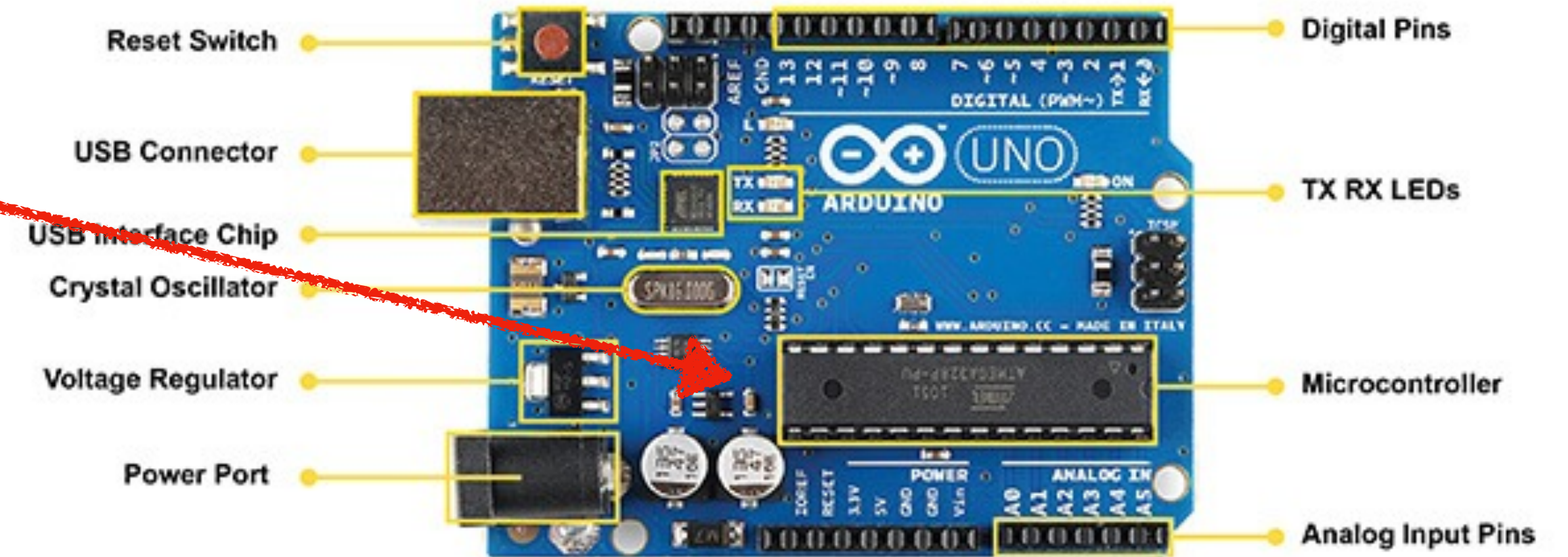


# Components of an Arduino board



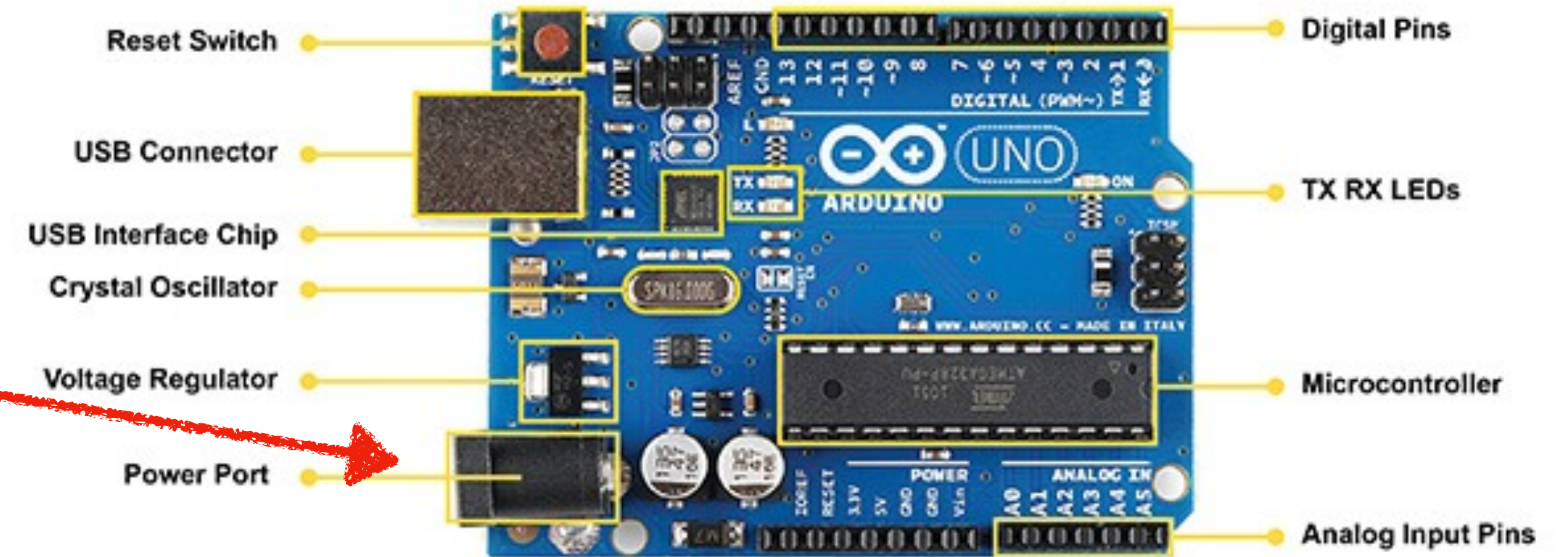
# Arduino Uno board anatomy

- Microcontroller unit (MCU)
- Power connector
- USB connector



# Arduino Uno board anatomy

- Microcontroller unit (MCU)
- Power connector
- USB connector



# Arduino Uno board anatomy

- Microcontroller unit (MCU)
- Power connector
- USB connector



# Arduino Uno board anatomy

- Analog and digital pins
- PWM pins
- I2C, SPI, and UART communication pins



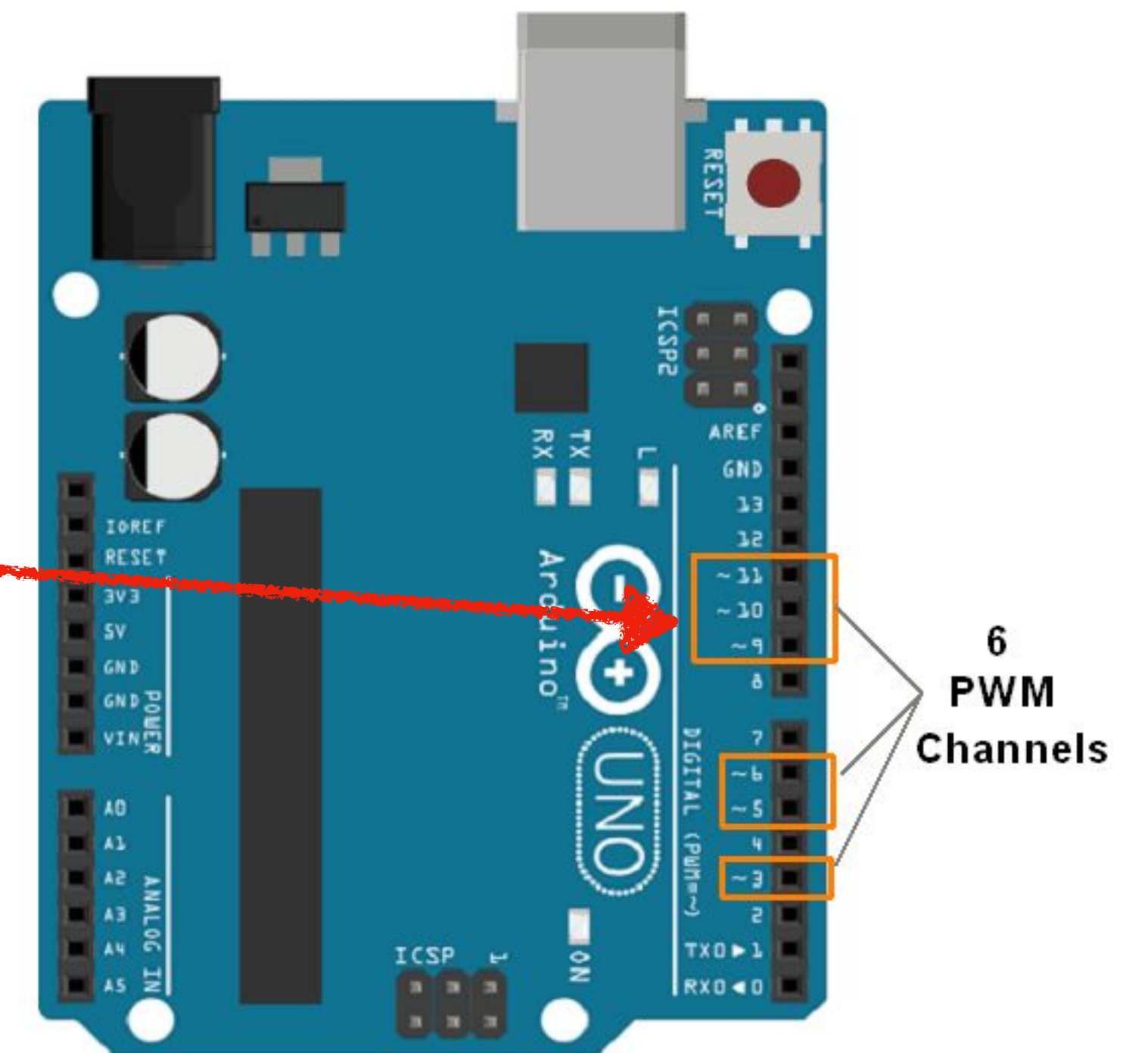
# Arduino Uno board anatomy

- Analog and digital pins
- PWM pins
- I2C, SPI, and UART communication pins



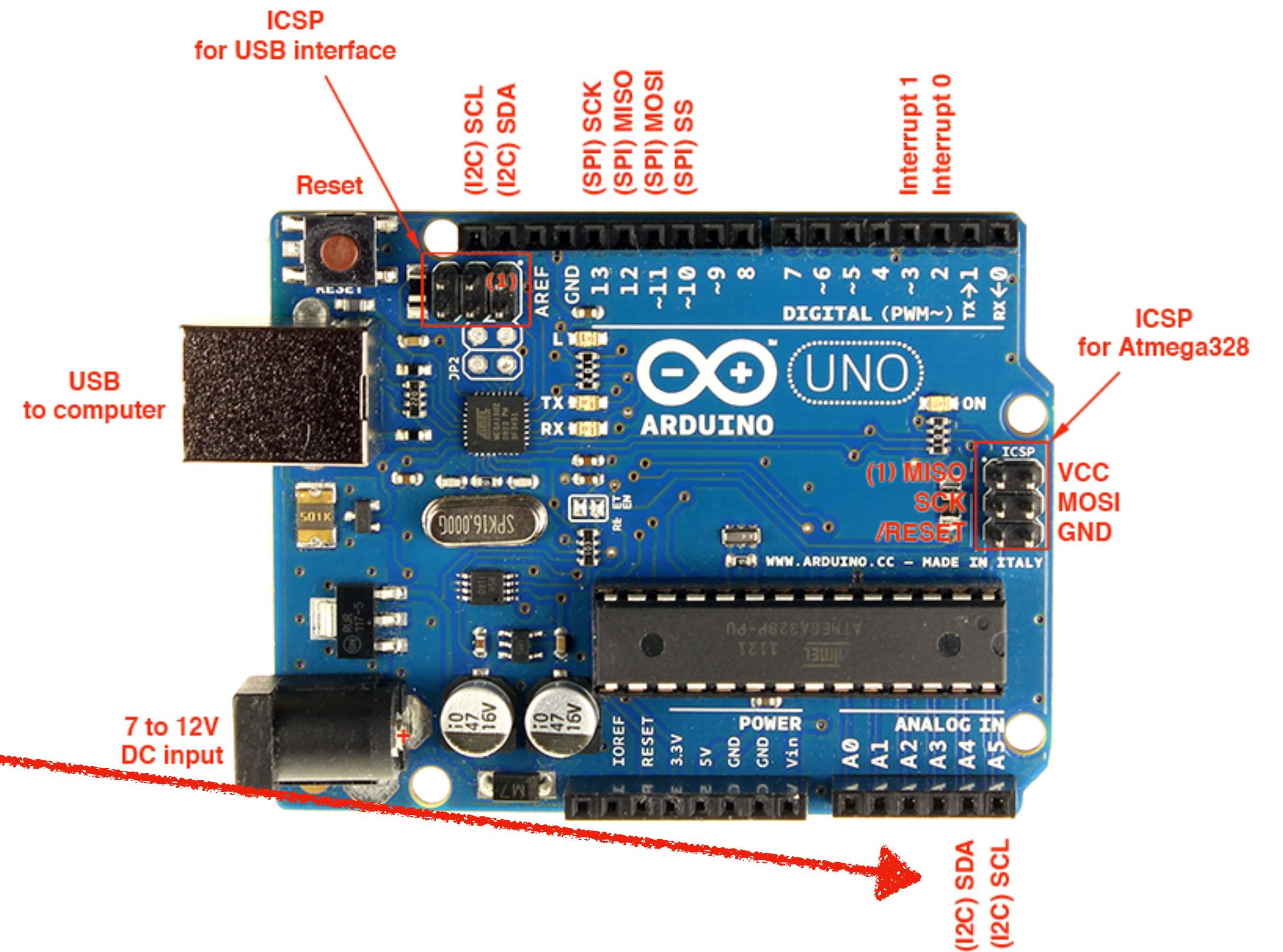
# Arduino Uno board anatomy

- Analog and digital pins
- PWM pins
- I2C, SPI, and UART communication pins



# Arduino Uno board anatomy

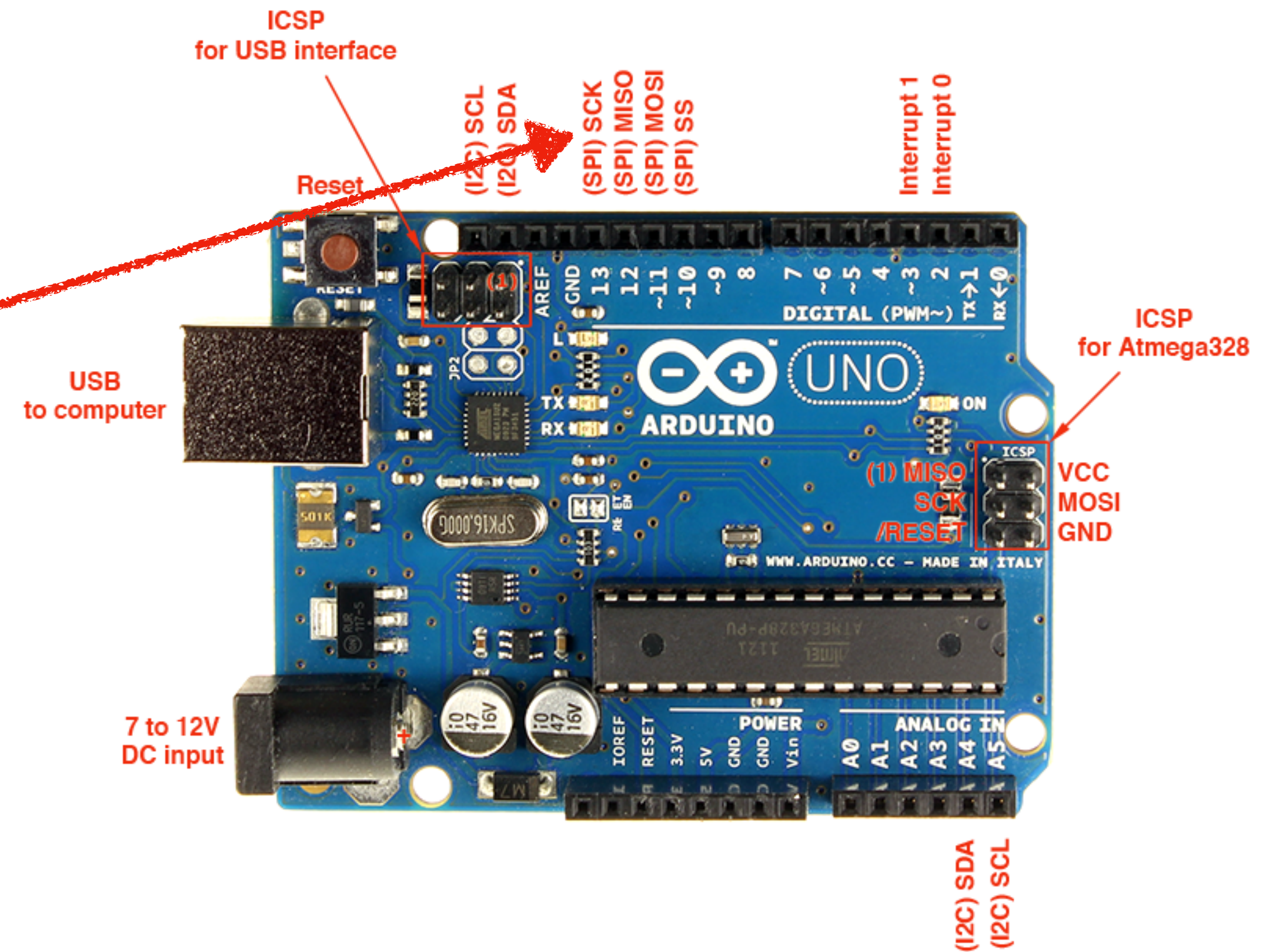
- Analog and digital pins
- PWM pins
- I2C, SPI, and UART communication pins





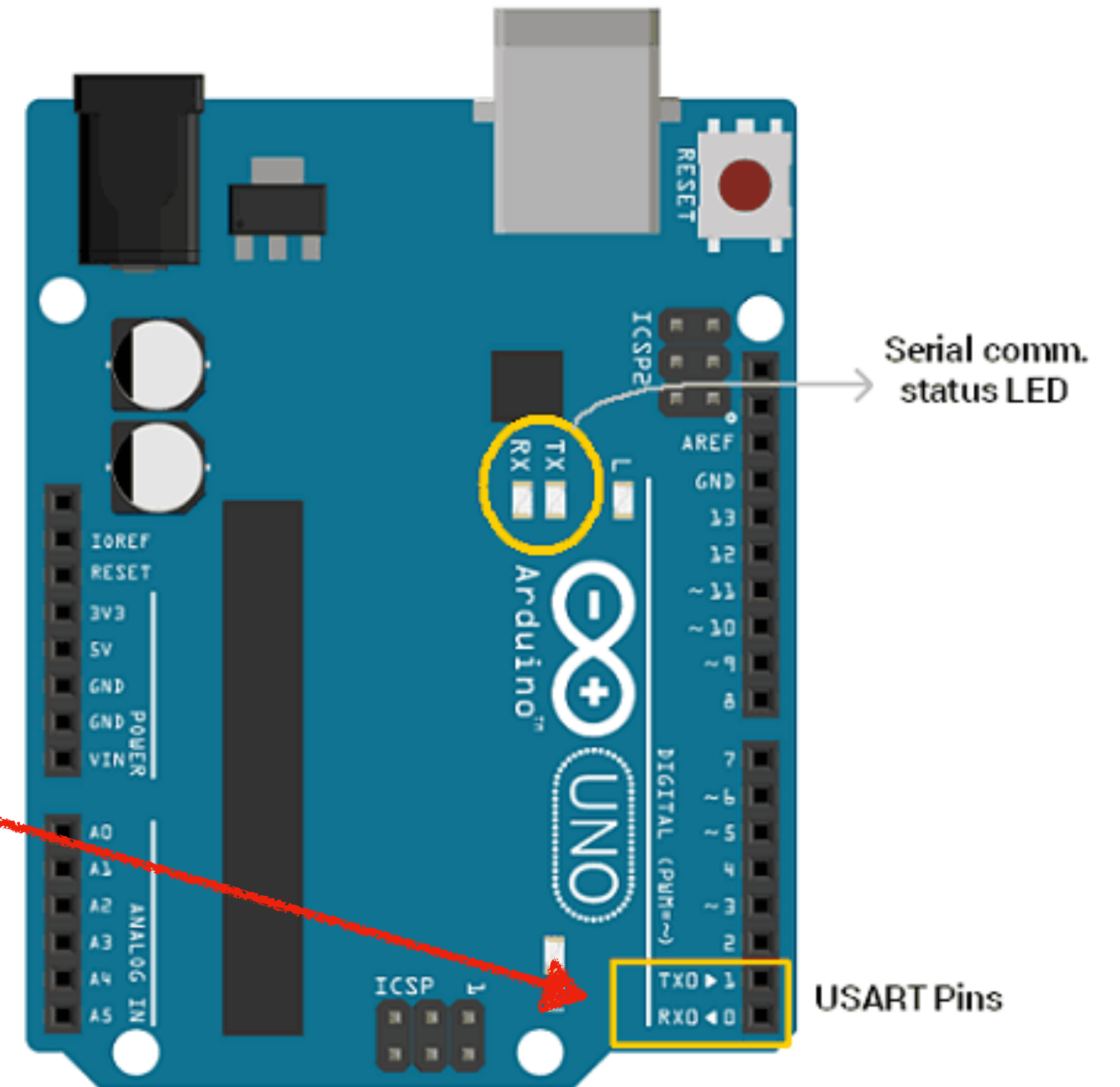
# Arduino Uno board anatomy

- Analog and digital pins
- PWM pins
- I2C, SPI, and UART communication pins



# Arduino Uno board anatomy

- Analog and digital pins
- PWM pins
- I2C, SPI, and UART communication pins



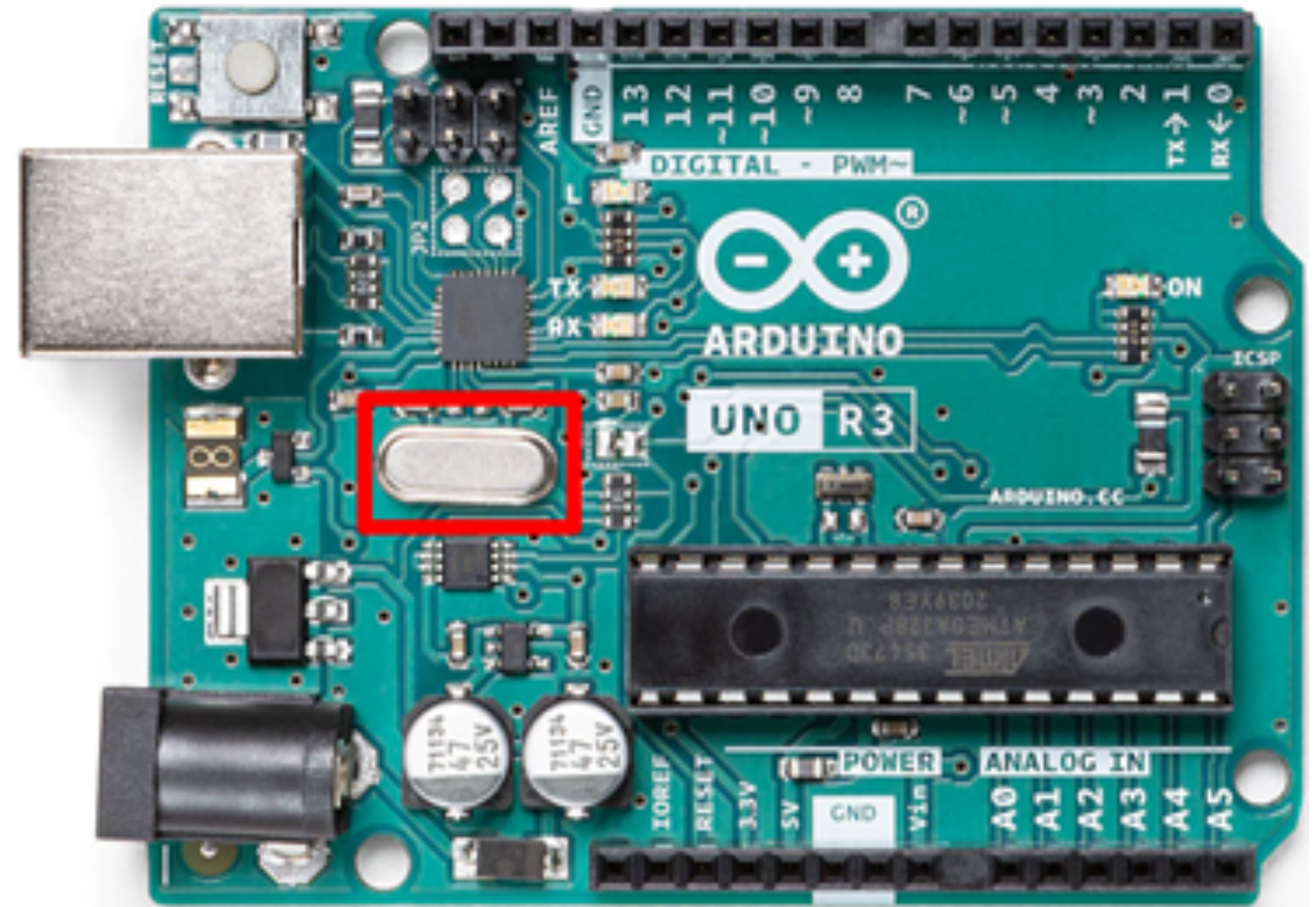
# Arduino Uno board anatomy

- Atmega328P MCU
- Flash memory
- EEPROM memory
- Clock



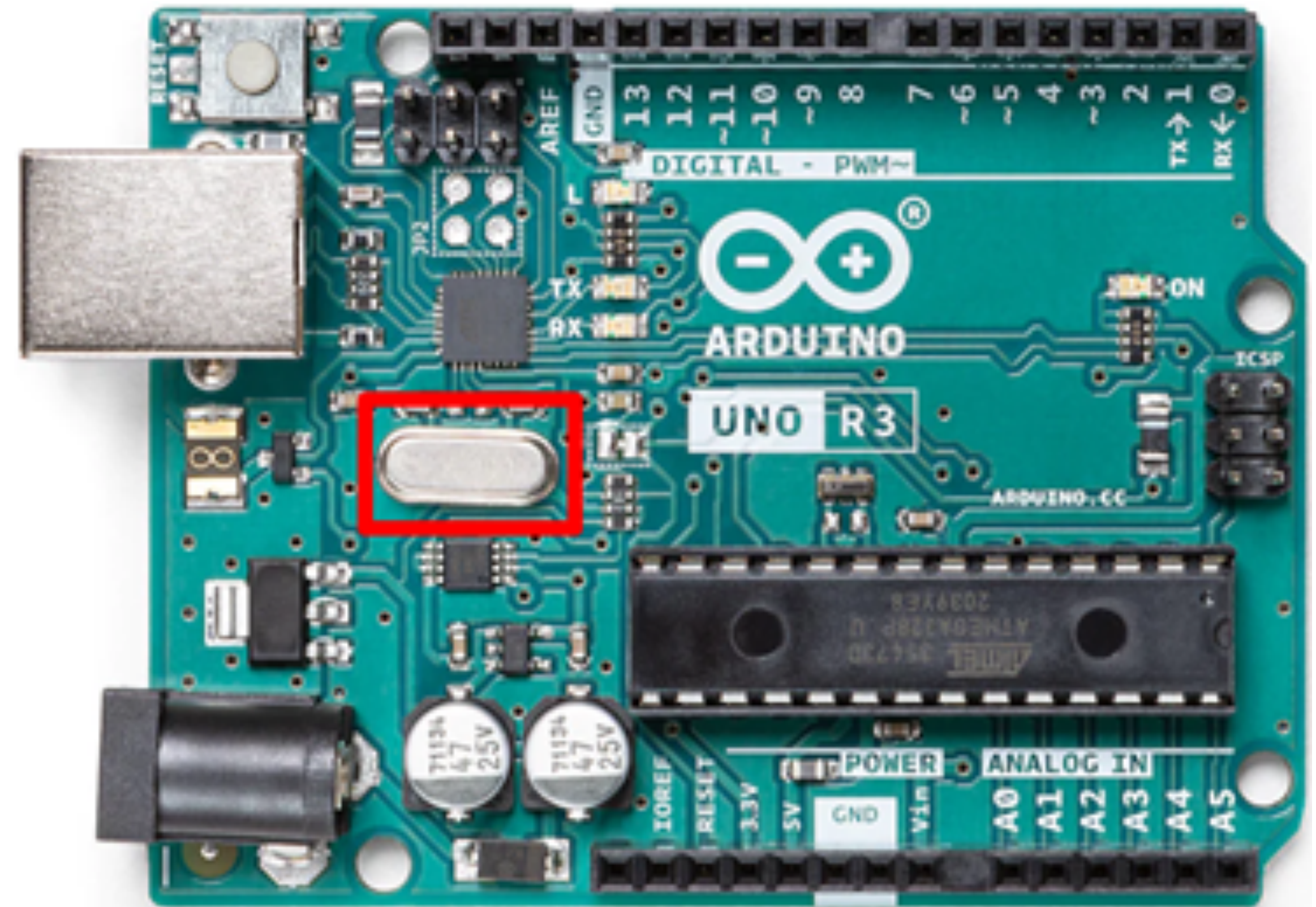
# Arduino Uno board anatomy

- Atmega328P MCU
- Flash memory
- EEPROM memory
- Clock - Crystal Oscillator



# Arduino Uno board anatomy

Arduino Board	Oscillator Frequency
Uno	16MHz
Nano	16MHz
MKR Wi-Fi	48MHz
Mega2560	16MHz
Due	84MHz



# The Arduino IDE

The image shows a screenshot of the Arduino IDE 2.0.0-rc9 interface. The window title is "AnalogReadSerial | Arduino IDE 2.0.0-rc9". The top toolbar contains several icons: a checkmark (Verify/Upload), a right arrow (Select Board & Port), a play button with a bug (Open Serial Monitor), a waveform (Open Serial Plotter), and a magnifying glass (Search). The main editor area displays the code for "AnalogReadSerial.ino". The code includes comments and a setup function. The left sidebar contains icons for the Sketchbook, Board Manager, Library Manager, Debugger, and Search. Callout boxes with lines pointing to these icons and toolbar buttons label them: "VERIFY/UPLOAD", "SELECT BOARD & PORT", "OPEN SERIAL MONITOR", "SKETCHBOOK", "BOARD MANAGER", "LIBRARY MANAGER", "DEBUGGER", "SEARCH", and "OPEN SERIAL PLOTTER".

**VERIFY/UPLOAD**      **SELECT BOARD & PORT**      **OPEN SERIAL MONITOR**

AnalogReadSerial | Arduino IDE 2.0.0-rc9

Arduino MKR WiFi 1010

**SKETCHBOOK**

```
1  /*
2   AnalogReadSerial
3   Reads an analog input on pin 0, prints the result to the Serial Monitor.
4   Graphical representation is available using Serial Plotter (Tools > Serial Plotter menu).
5   Connect a potentiometer to pin A0, and the outside pins to +5V and ground.
6   This example code is in the public domain.
7   https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogReadSerial
8   */
9
10 void setup() {
11   // initialize serial communication at 9600 bits per second:
12   Serial.begin(9600);
13 }
14
15
16
17
18
```

**BOARD MANAGER**

**LIBRARY MANAGER**

**DEBUGGER**

**SEARCH**

**OPEN SERIAL PLOTTER**

# Hello World!

```
void setup() {  
  // initialize serial communication at 9600 baud:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // print "Hello, World!" to the serial monitor:  
  Serial.println("Hello, World!");  
  delay(1000); // wait for 1 second before repeating  
}
```

# Debugging and Troubleshooting

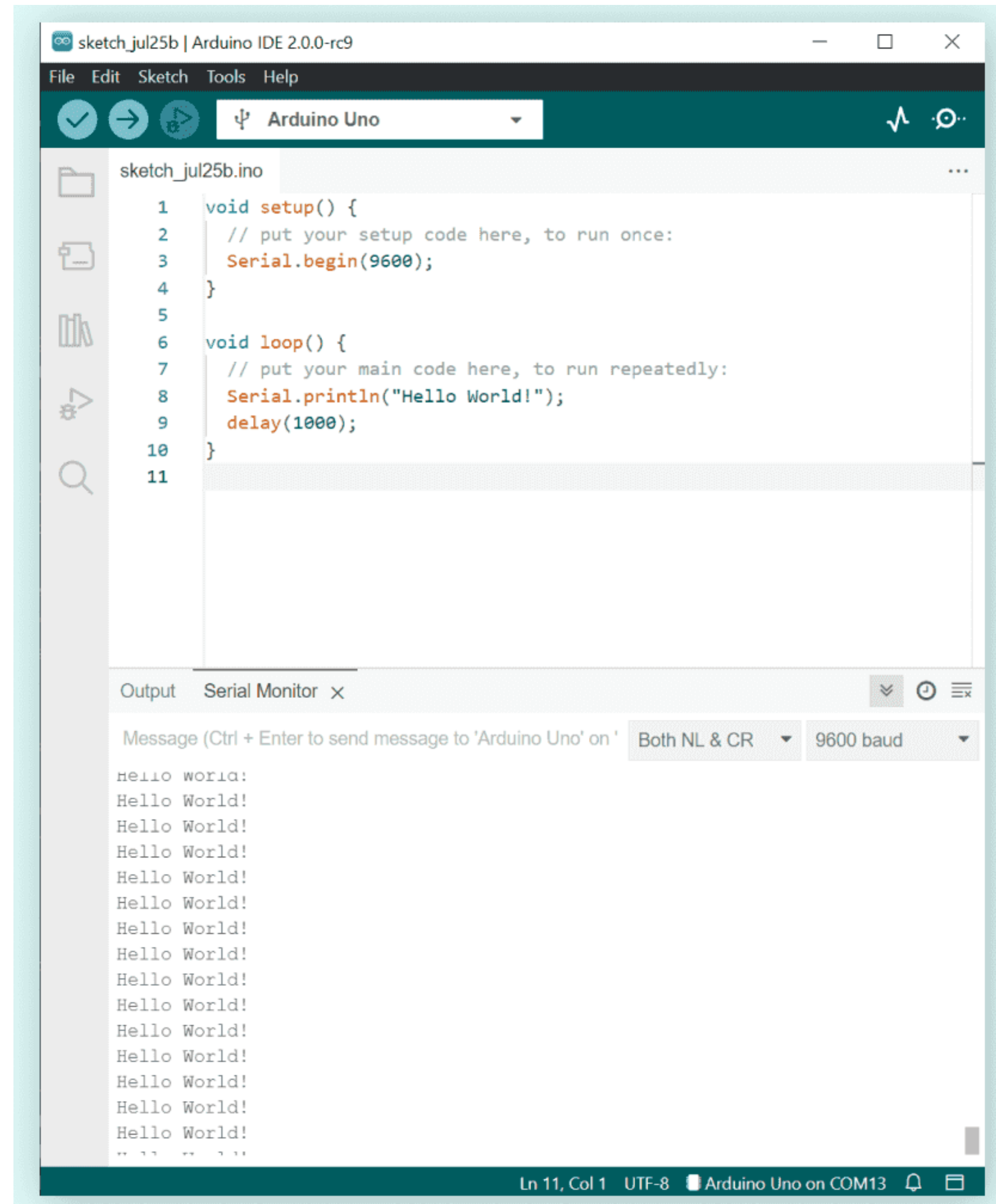
```
void setup() {  
  // initialize serial communication at 9600 baud:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // print "Hello, World!" to the serial monitor:  
  Serial.println("Hello, World!");  
  delay(1000); // wait for 1 second before repeating  
}
```



# Debugging and Troubleshooting

```
void setup() {  
  // initialize serial communication at 9600 baud:  
  Serial.begin(9600);  
}
```

```
void loop() {  
  // print "Hello, World!" to the serial monitor:  
  Serial.println("Hello, World!");  
  delay(1000); // wait for 1 second before repeating  
}
```



# Language

- Functions
- Variables
- Structure

Sketch	Arithmetic Operators	Pointer Access Operators	ifiers Bytes
loop()	oint Constants % (remainder)	& (reference operator)	
setup()	onstants * (multiplication)	* (dereference operator)	)
Control Structure	+ (addition)	Bitwise Operators	)
break	INPUT_ - (subtraction)	& (bitwise and)	)
continue	/ (division)	<< (bitshift left)	:()
do...while	= (assignment operator)	>> (bitshift right)	)
else	Comparison Operators	^ (bitwise xor)	nterrupts
for	!= (not equal to)	(bitwise or)	attachInterrupt()
goto	< (less than)	~ (bitwise not)	interrupt()
if	<= (less than or equal to)	Compound Operators	ts
return	== (equal to)	%= (compound remainder)	ts()
switch...case	> (greater than)	&= (compound bitwise and)	rupts()
while	>= (greater than or equal to)	*= (compound multiplication)	ication
Further Syntax	Boolean Operators	++ (increment)	
#define (define)	! (logical not)	+= (compound addition)	
#include (include)	&& (logical and)	-- (decrement)	
/* */ (block comment)	(logical or)	-= (compound subtraction)	
// (single line comment)		/= (compound division)	
;		^= (compound bitwise xor)	d
{}		= (compound bitwise or)	

# Accessing the pins

```
int ledPin = 13; // LED connected to digital pin 13
int inPin = 7;   // pushbutton connected to digital pin 7
int val = 0;    // variable to store the read value

void setup() {
  pinMode(ledPin, OUTPUT); // sets the digital pin 13 as output
  pinMode(inPin, INPUT);  // sets the digital pin 7 as input
}

void loop() {
  val = digitalRead(inPin); // read the input pin
  digitalWrite(ledPin, val); // sets the LED to the button's value
}
```

# Accessing the pins

```
int analogPin = A3; // potentiometer wiper (middle terminal) connected to analog pin 3
                    // outside leads to ground and +5V
int val = 0; // variable to store the value read

void setup() {
  Serial.begin(9600); // setup serial
}

void loop() {
  val = analogRead(analogPin); // read the input pin
  Serial.println(val); // debug value
}
```

# Accessing the pins

```
int ledPin = 9;    // LED connected to digital pin 9
int analogPin = 3; // potentiometer connected to analog pin 3
int val = 0;      // variable to store the read value

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

void loop() {
  val = analogRead(analogPin); // read the input pin
  analogWrite(ledPin, val / 4);
  // analogRead values go from 0 to 1023, analogWrite values from 0 to 255
}
```

# Blink

```
// Define the pin number
int ledPin = 13;

void setup() {
  // Set the LED pin as an output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // Turn the LED on
  digitalWrite(ledPin, HIGH);
  // Wait for 1 second
  delay(1000);
  // Turn the LED off
  digitalWrite(ledPin, LOW);
  // Wait for another second
  delay(1000);
}
```

# Projects using Arduino

- Smart Home Devices
  - Smart lighting system
  - Smart Thermostat
  - Home Security System

# Projects using Arduino

- Environmental Monitoring Systems
- Weather Station
- Air Quality Monitor
- Water Quality Monitor



# Projects using Arduino

- Industrial Automation
  - Inventory Control System
  - Automated Irrigation System
  - Conveyor Belt Controller

# Projects using Arduino

- Personal Health Monitoring
  - Fitness Tracker
  - Blood Pressure Monitor
  - Smart Pill Dispenser

# Lecture outcomes

- Using Arduino platform.
- Hello World! program.

