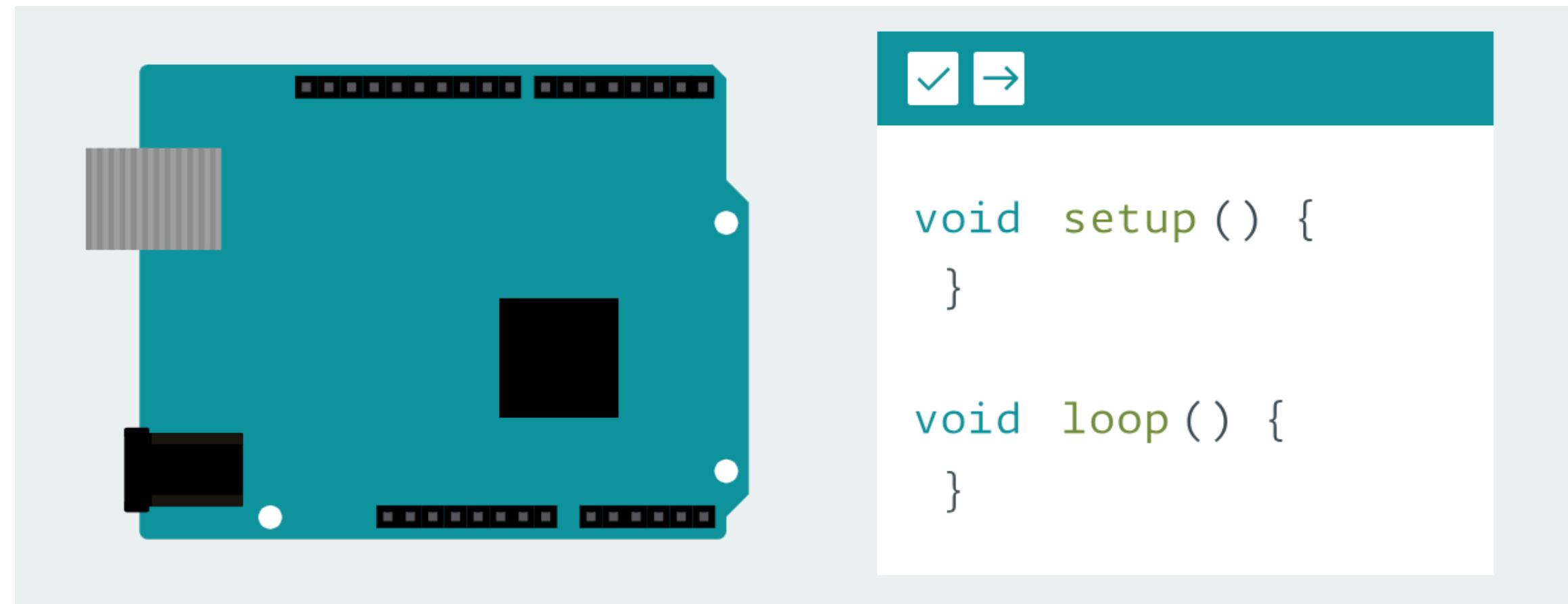


# Lecture #3

# Arduino

Android Things 2023

# Foundations



1972 1982

1999 2003

# History

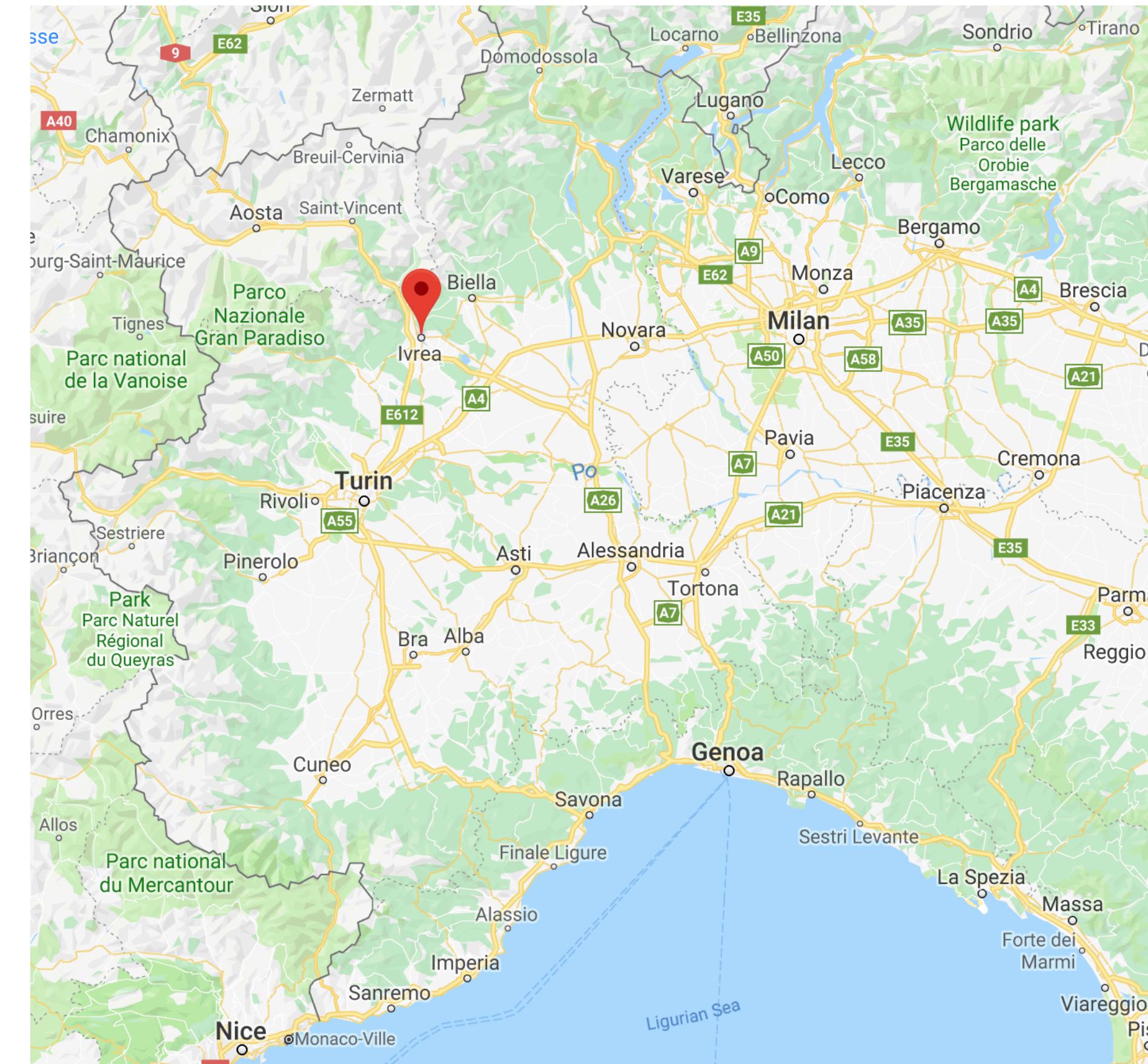


King of Italy (1002–1014)

# History



**Hernando Barragán**



Interaction Design Institute Ivrea  
IDII

# History



**Casey Reas**



**Massimo Banzi**



**2001**  
<https://processing.org/>

# History

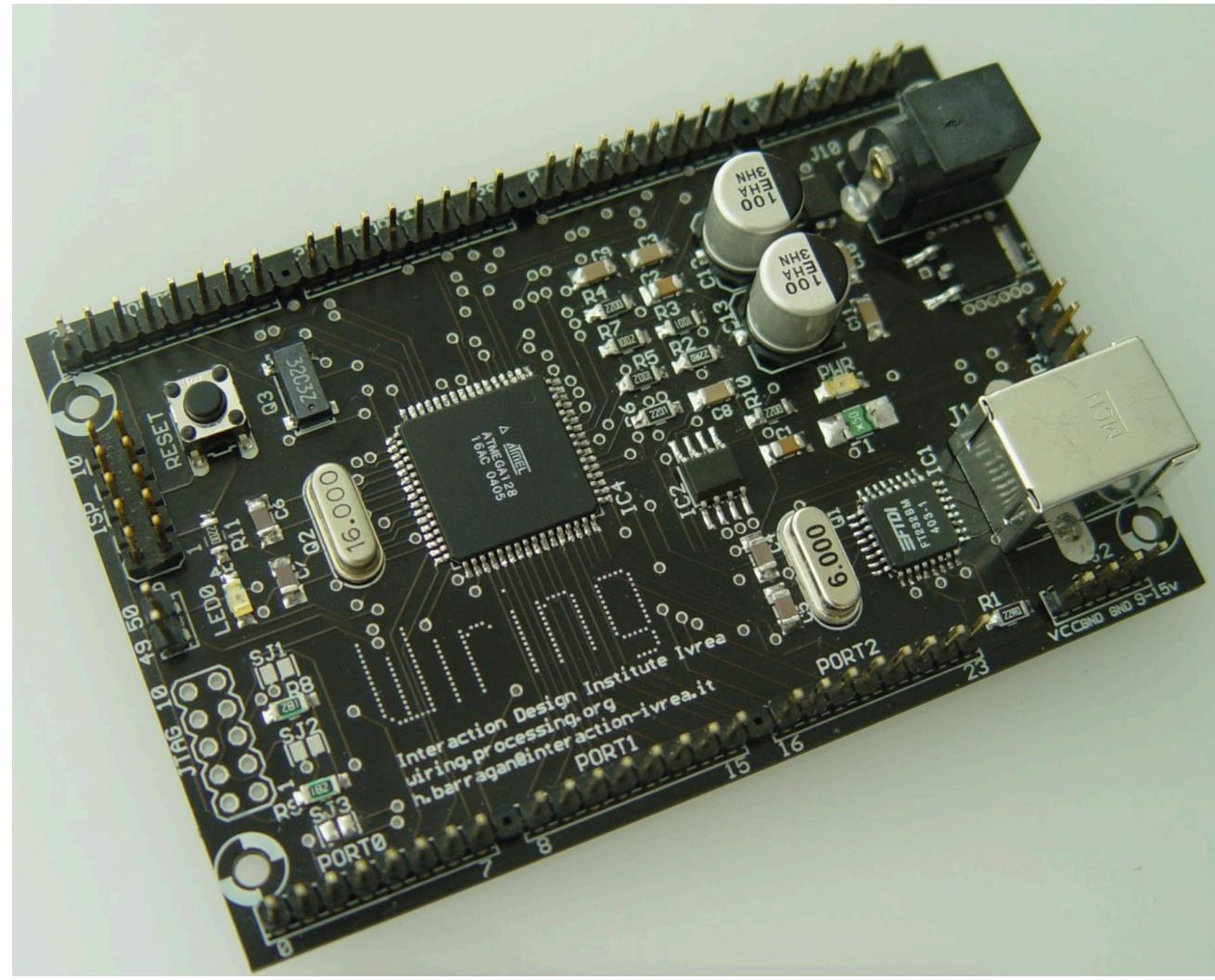


*Wiring*

2003

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

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
}
```

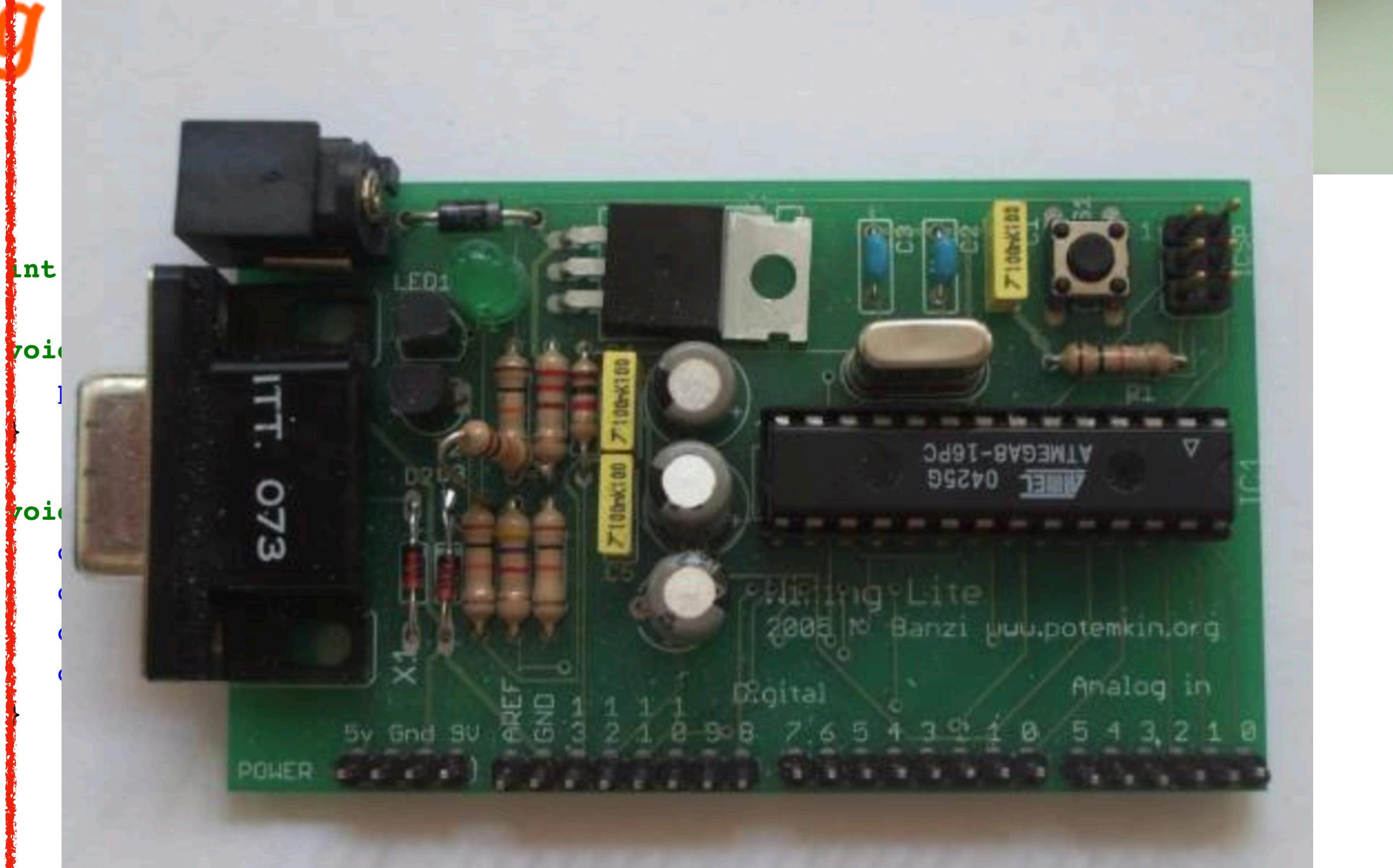
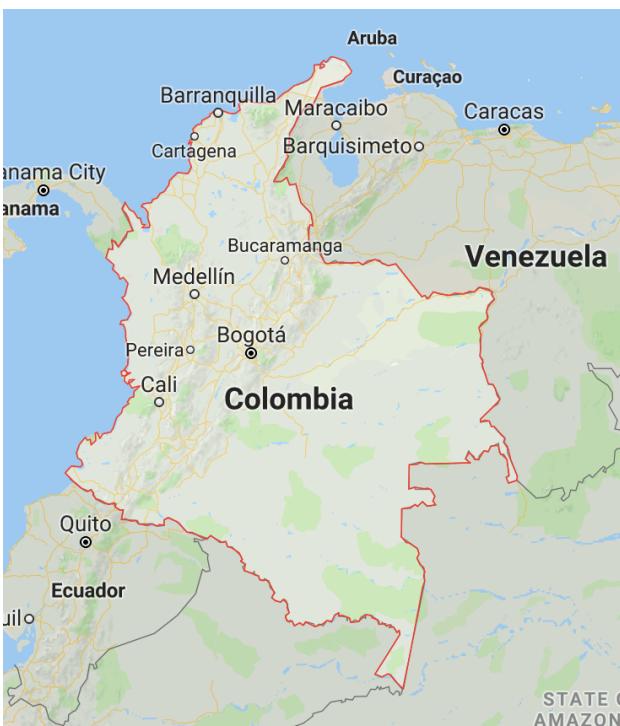


# History



Wiring *Wiring*

2003



2004

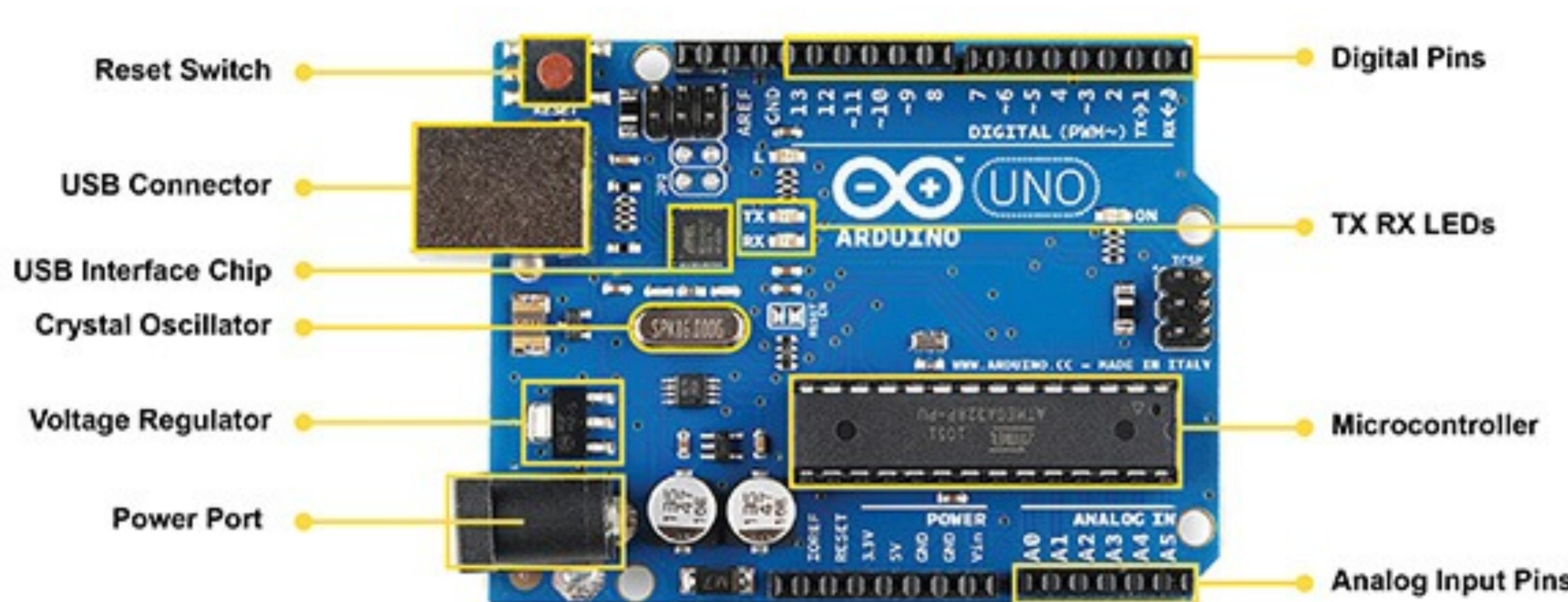
Arduino

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

# Credits

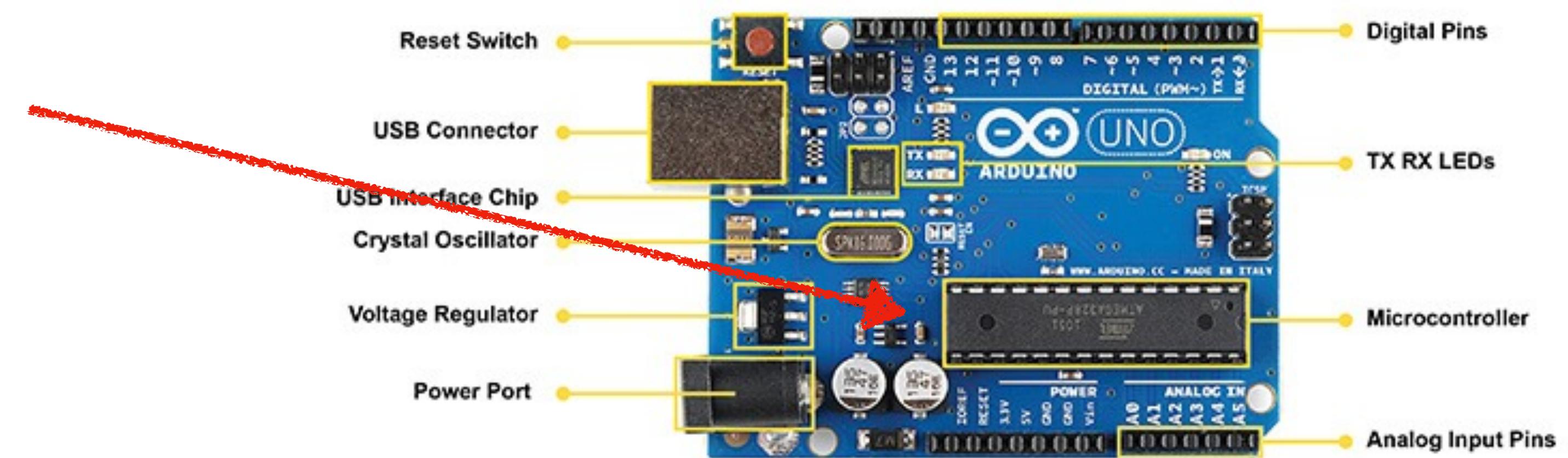
Originally started as a research project by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis at the Interaction Design Institute of Ivrea in the early 2000s, it builds upon the Processing project, a language for learning how to code within the context of the visual arts 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



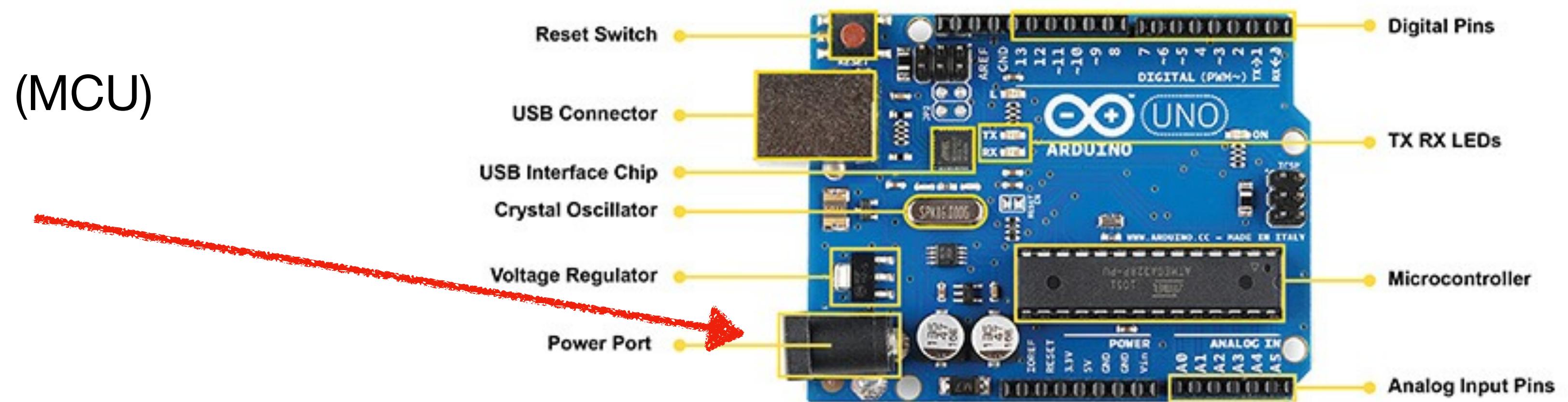
# Arduino Uno board anatomy

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



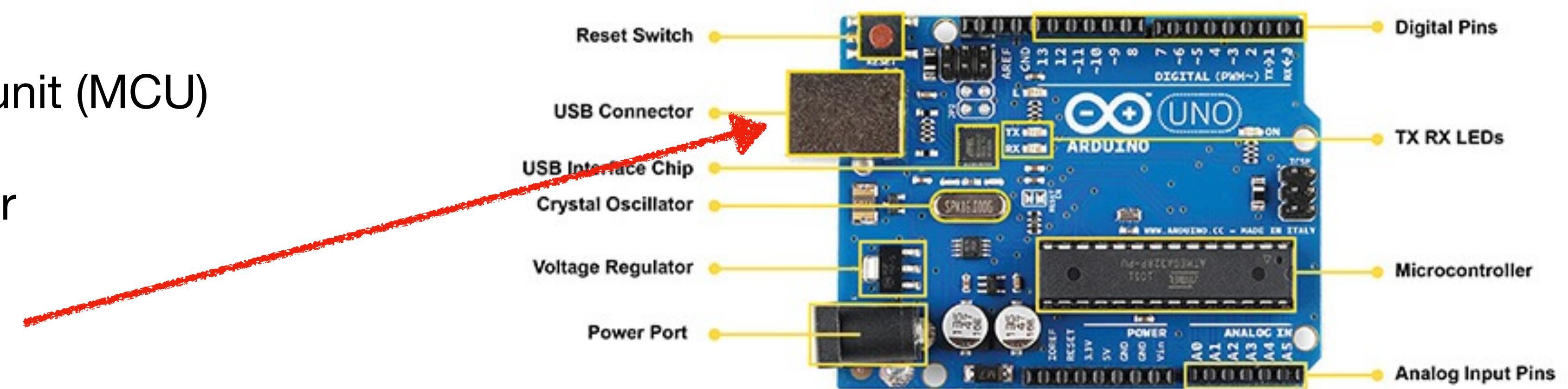
# Arduino Uno board anatomy

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



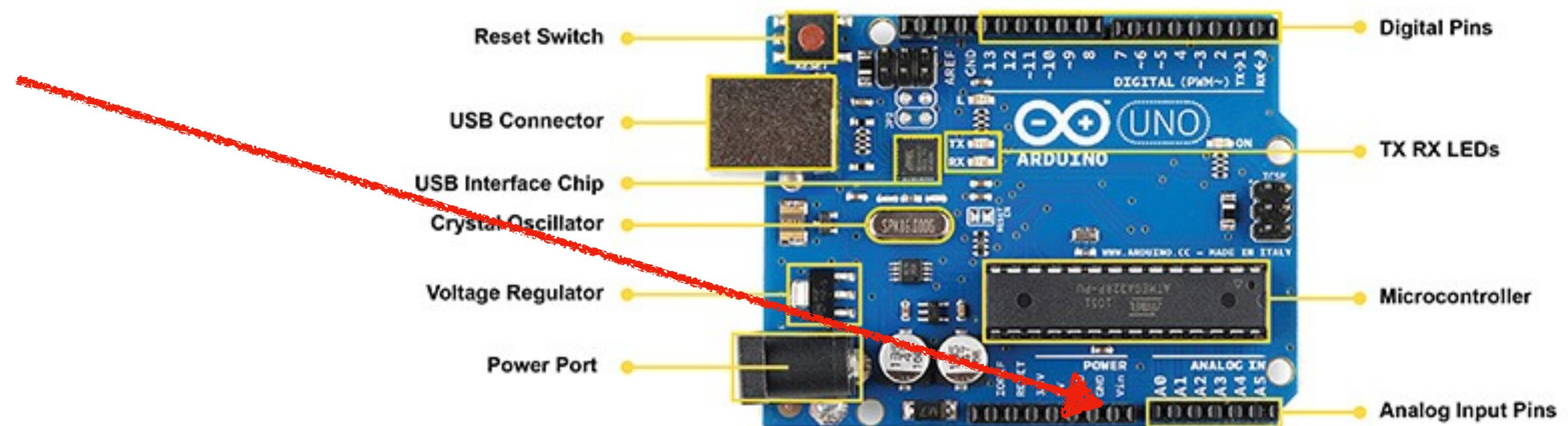
# Arduino Uno board anatomy

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



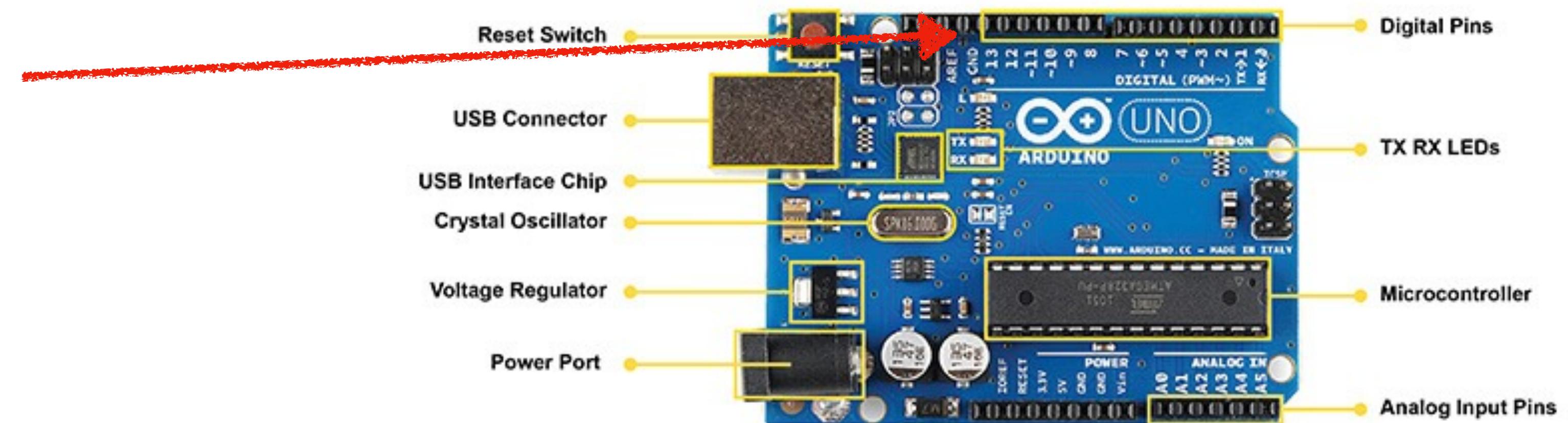
# Arduino Uno board anatomy

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



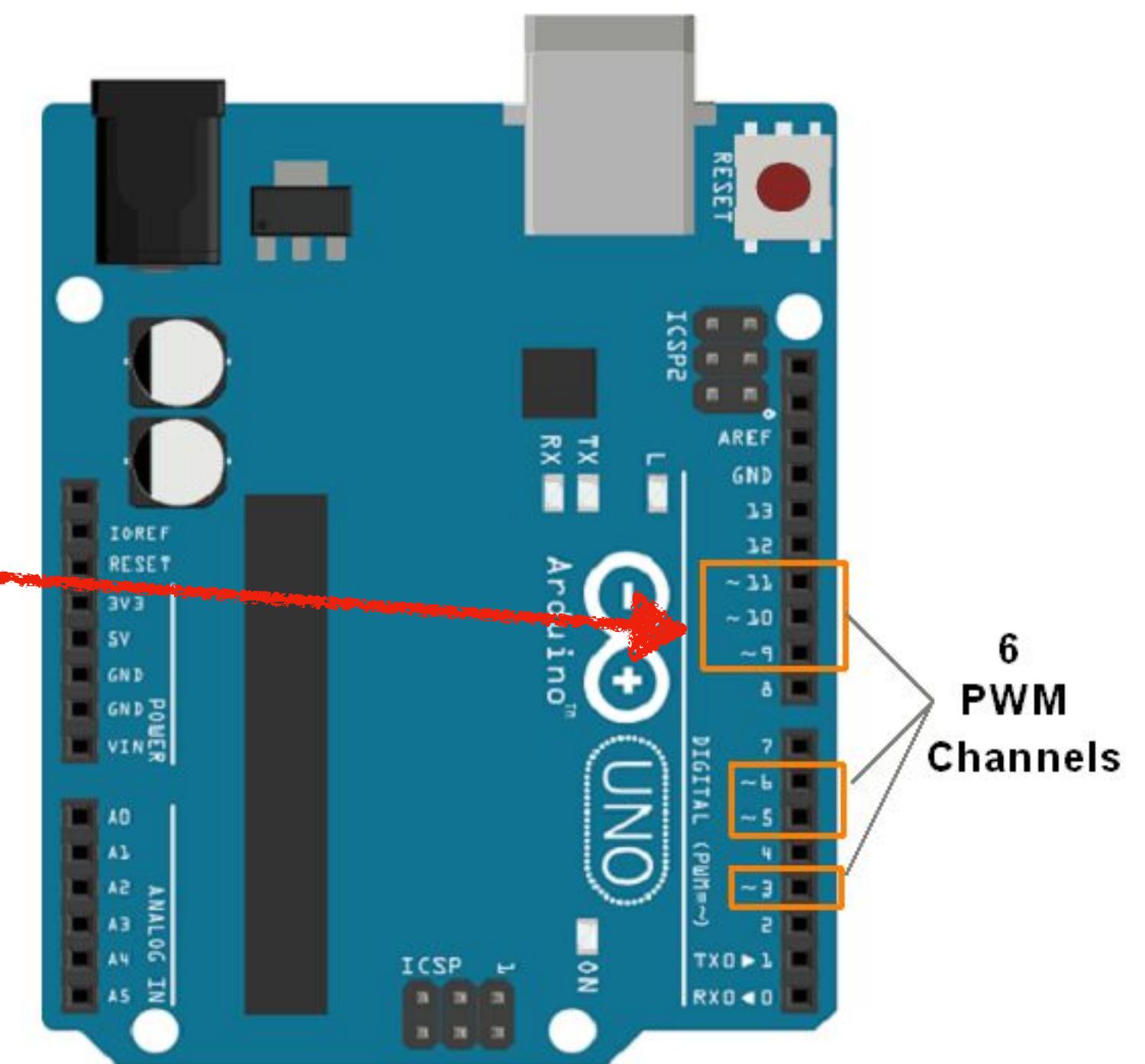
# Arduino Uno board anatomy

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



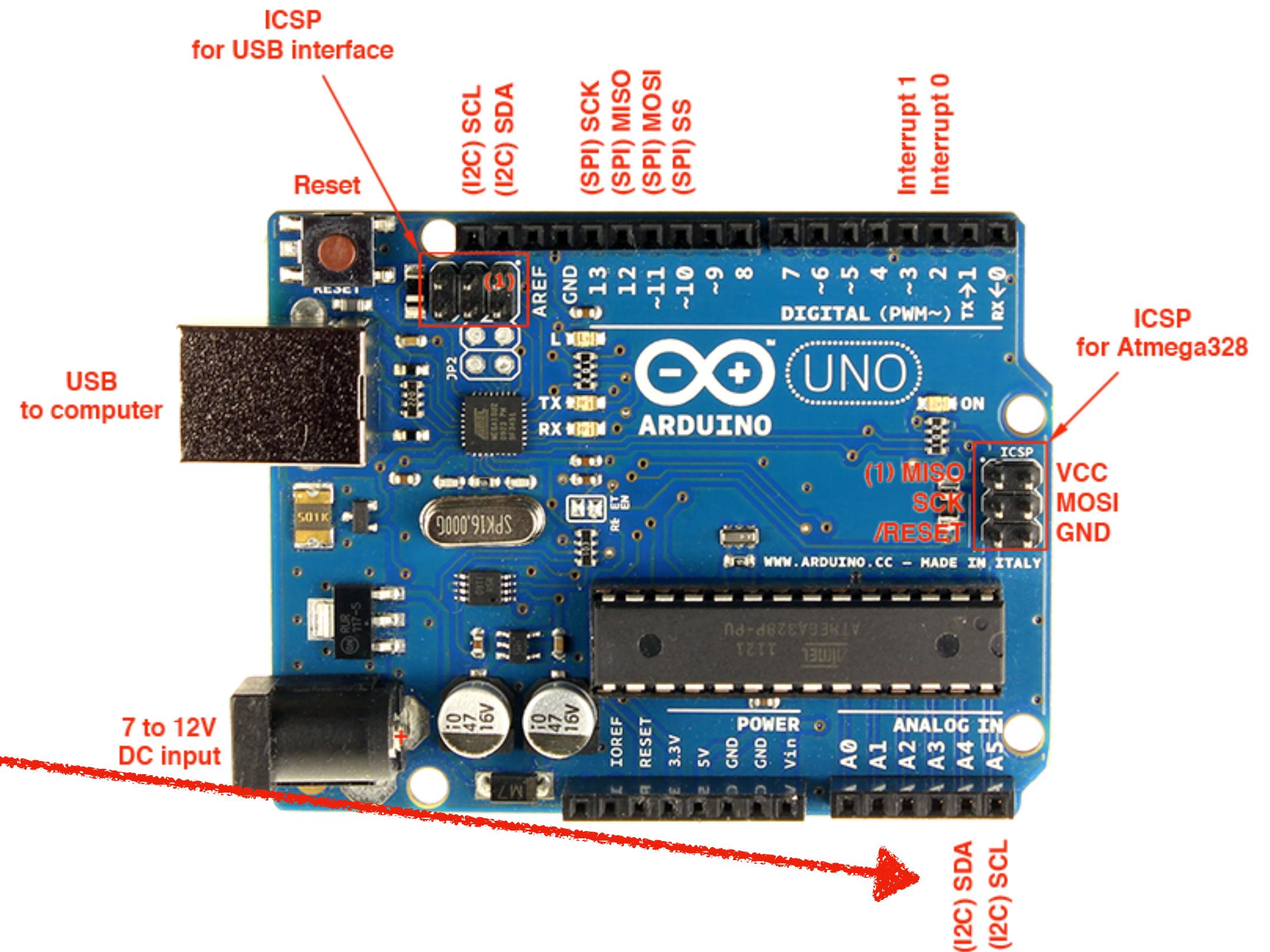
# Arduino Uno board anatomy

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



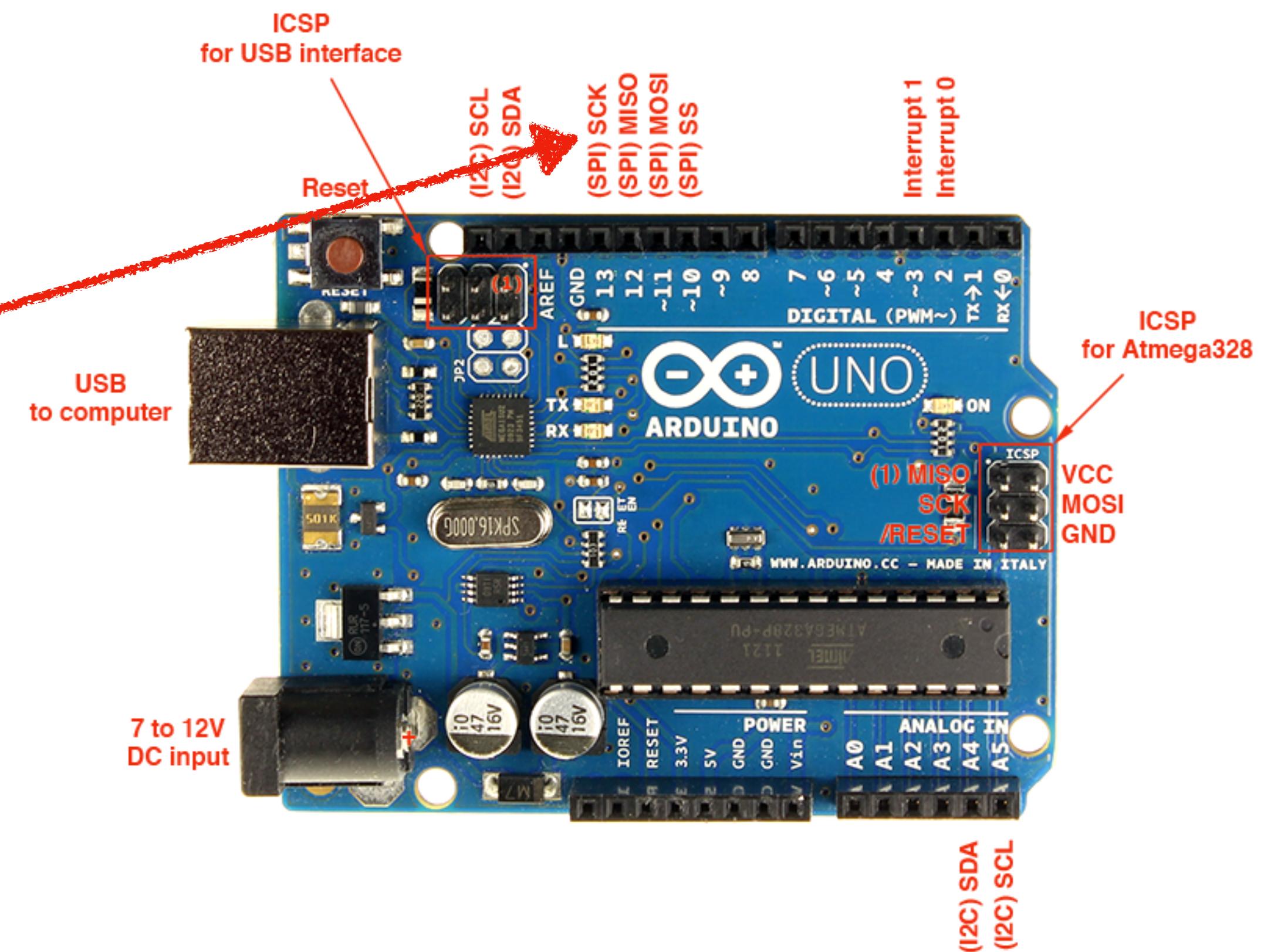
# Arduino Uno board anatomy

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



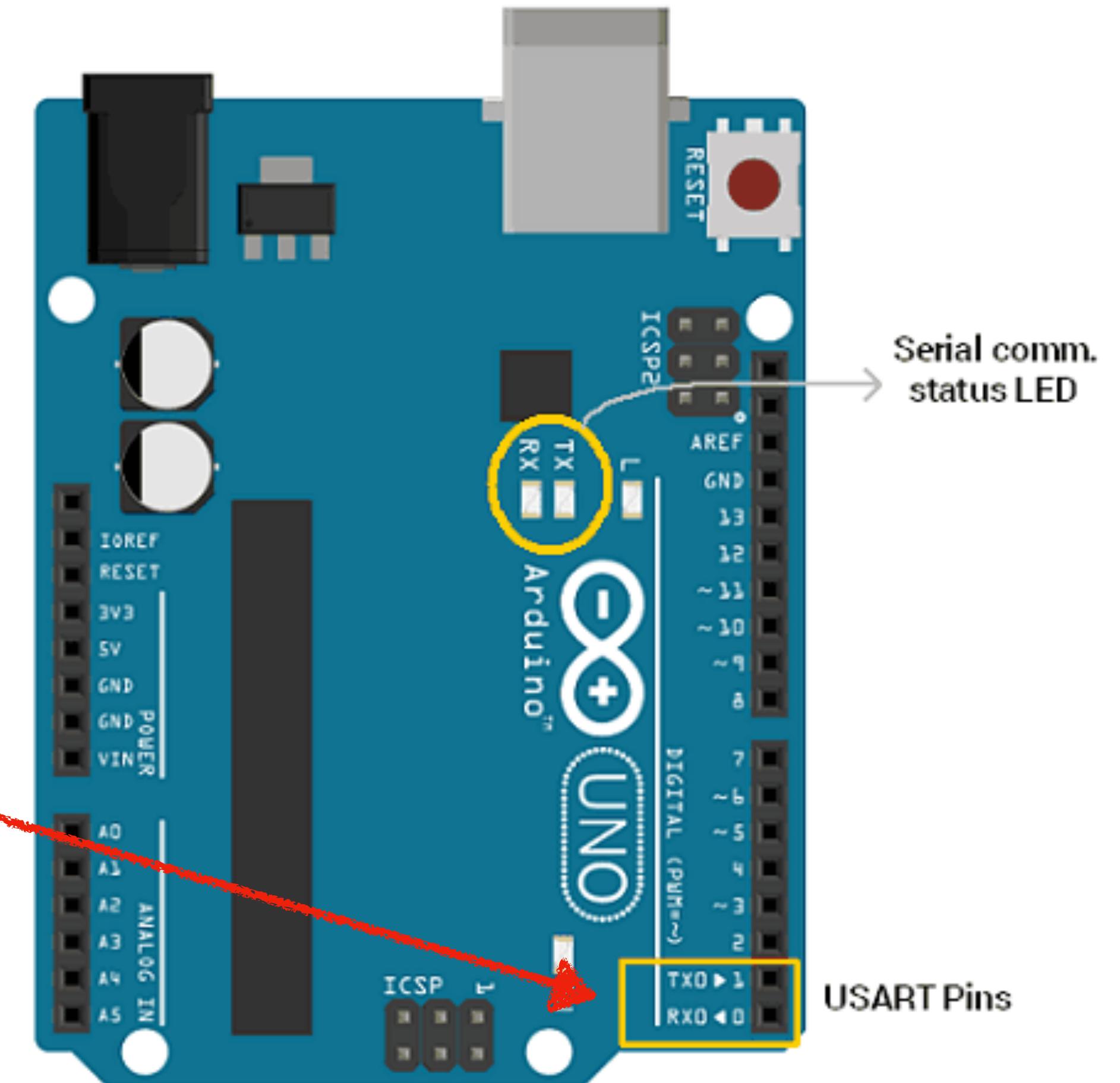
# Arduino Uno board anatomy

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



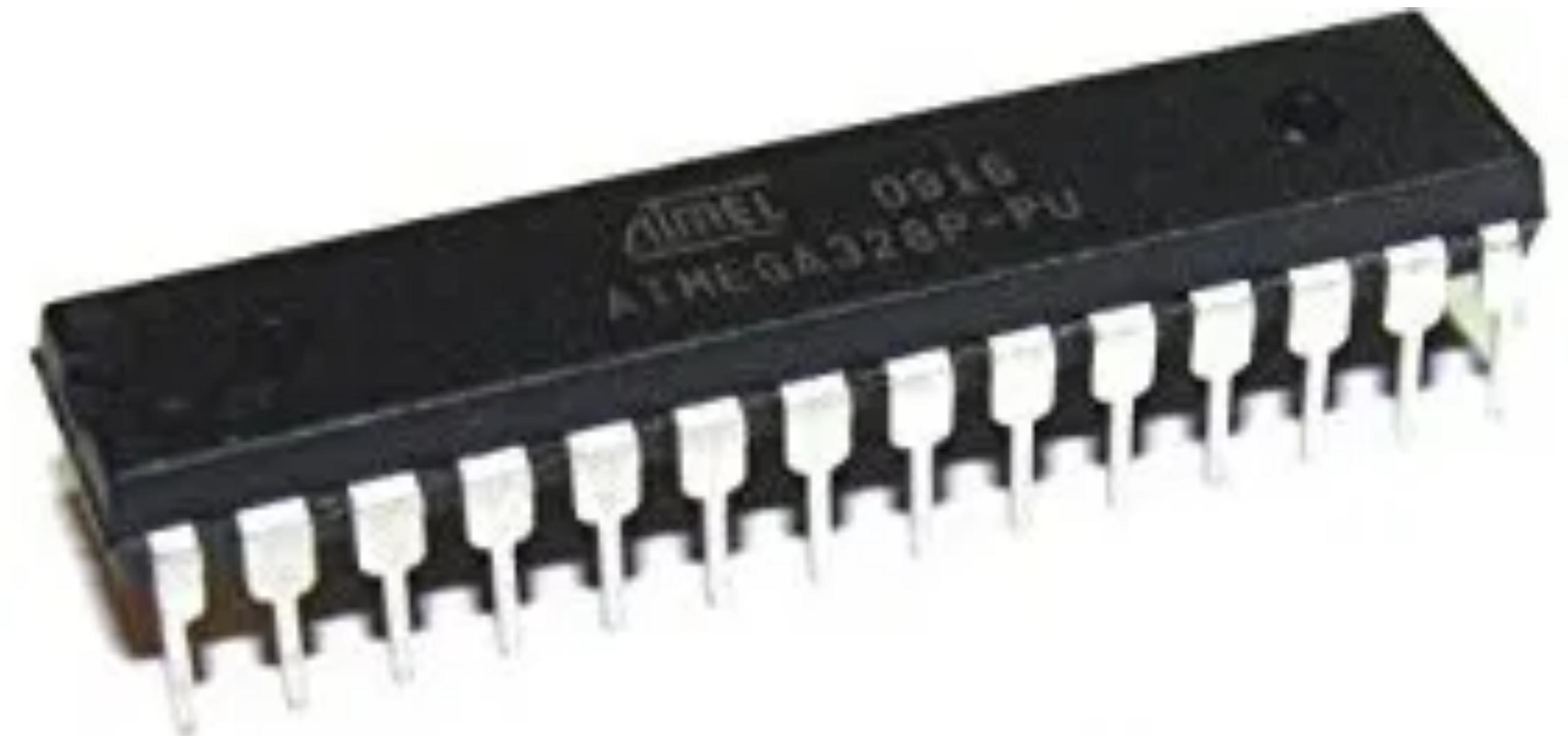
# Arduino Uno board anatomy

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



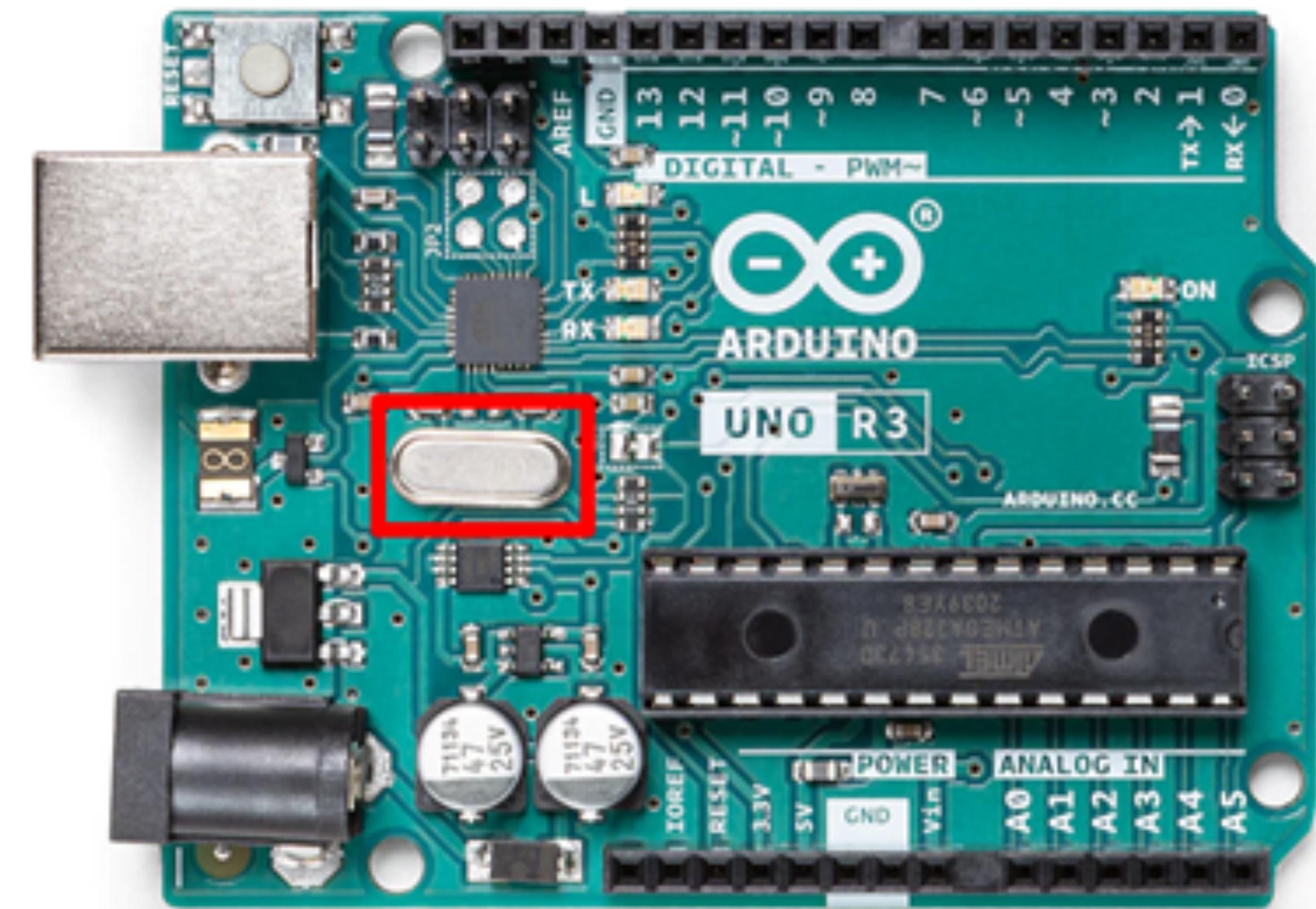
# Arduino Uno board anatomy

- Atmega328P MCU
- Flash memory
- EEPROM memory
- Clock



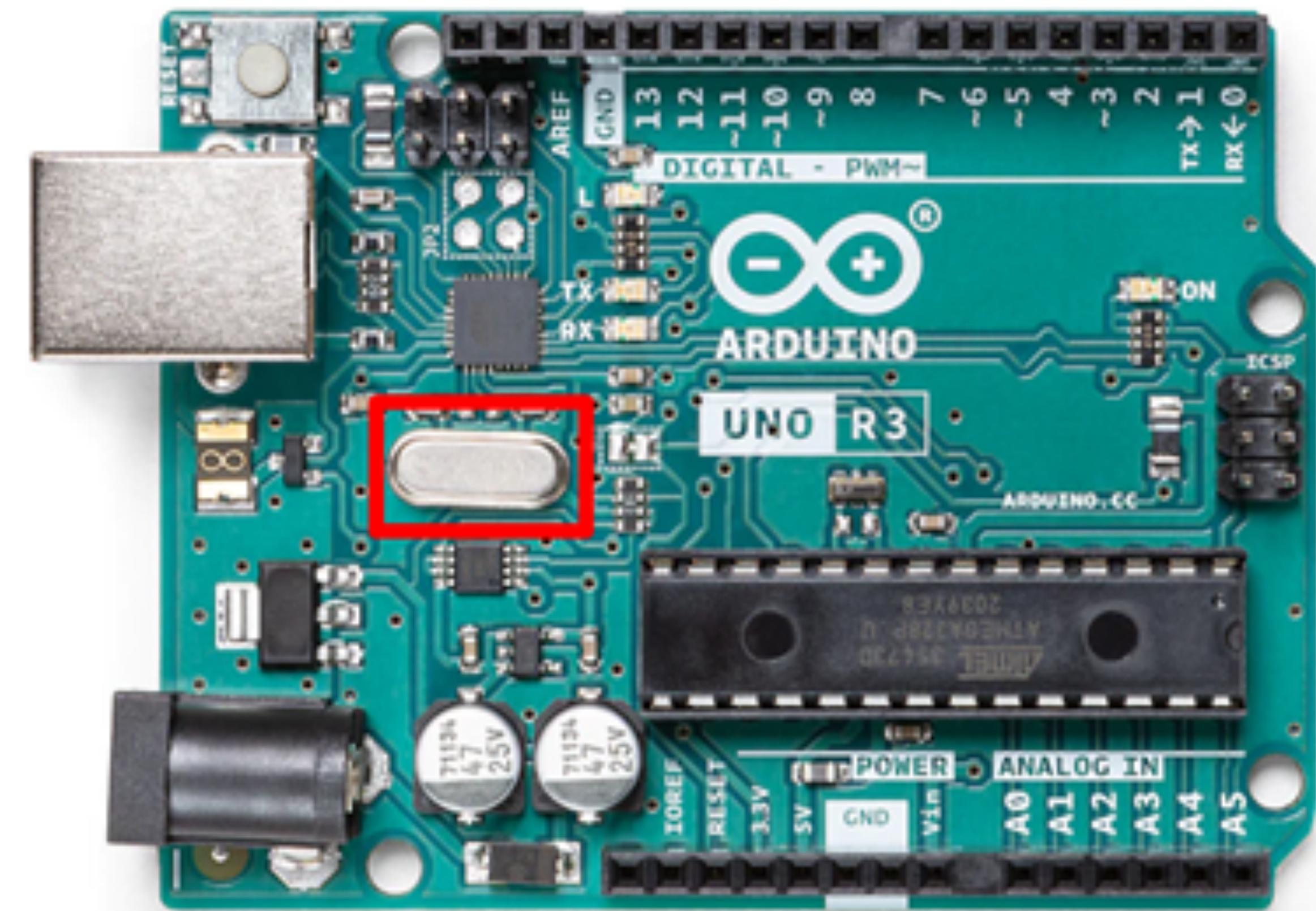
# Arduino Uno board anatomy

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

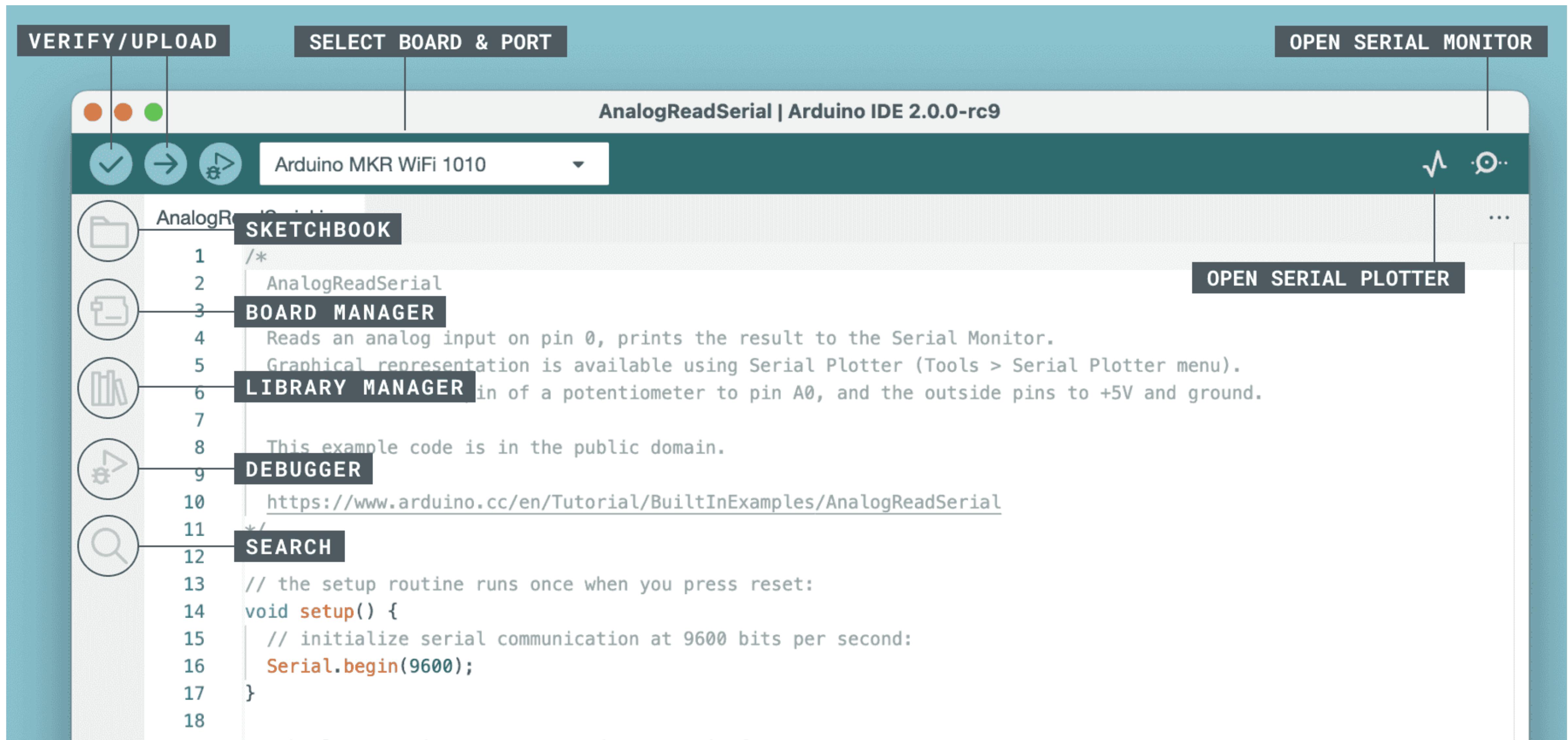


# Arduino Uno board anatomy

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



# The Arduino IDE



# 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  
}
```

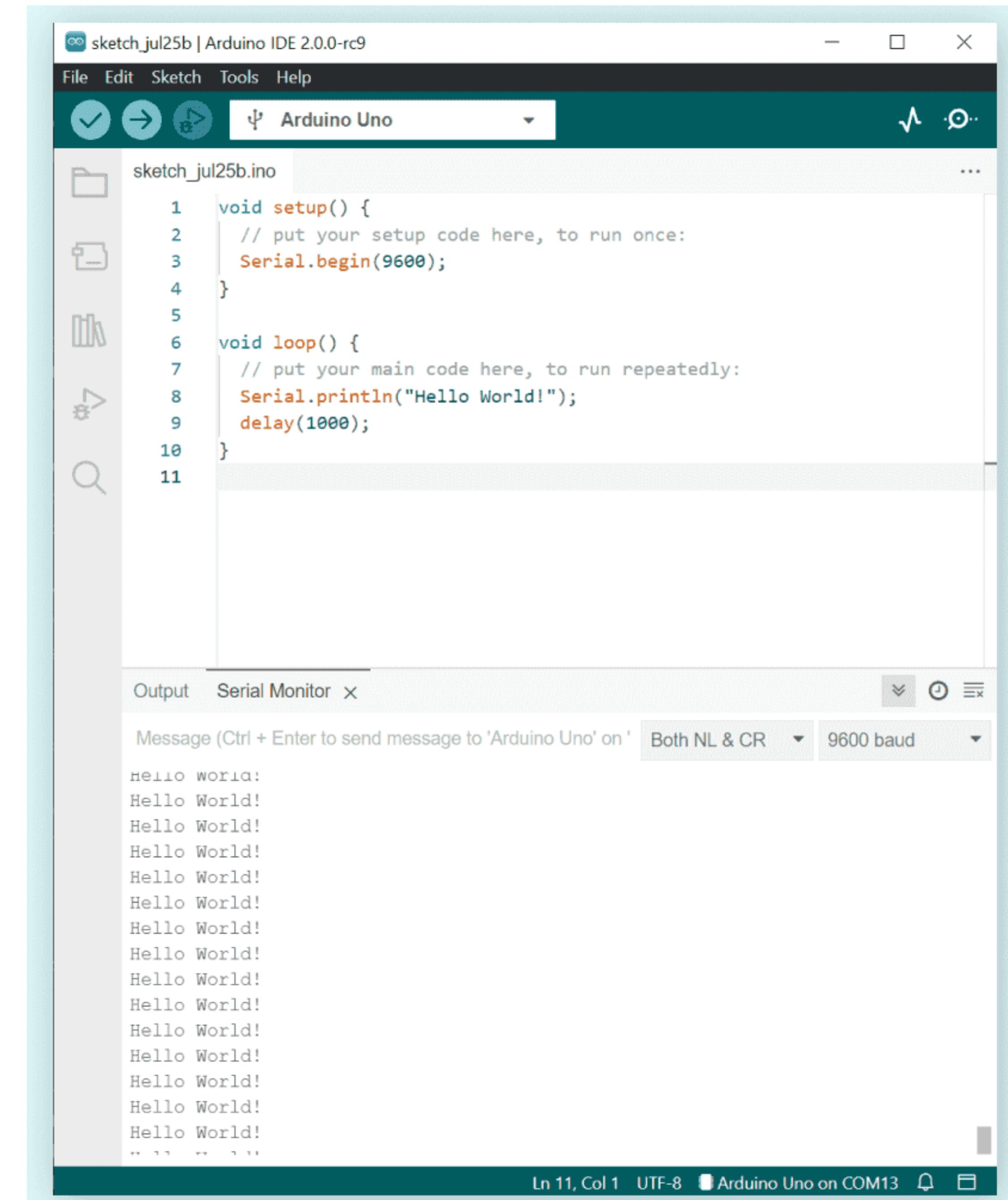
# Debugging and Troubleshooting

```
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	Keywords	Bytes
loop()	int Constants	% (remainder)	& (reference operator)	
setup()	Variables	*	* (dereference operator)	)
Control Structure		+		)
break	INPUT_	- (subtraction)	& (bitwise and)	
continue		/ (division)	<< (bitshift left)	)
do...while		= (assignment operator)	>> (bitshift right)	:0
else		Comparison Operators	^ (bitwise xor)	)
for		!= (not equal to)	(bitwise or)	nterrupts
goto	Family	< (less than)	~ (bitwise not)	attachInterrupt()
if	unction()	<= (less than or equal to)	Compound Operators	nterrupt()
return	unction()	== (equal to)	%= (compound remainder)	its
switch...case		> (greater than)	&= (compound bitwise and)	nts()
while		>= (greater than or equal to)	*= (compound multiplication)	rupts()
Further Syntax	Boolean Operators	int	++ (increment)	nication
#define (define)		long	+= (compound addition)	
#include (include)	!		-- (decrement)	
/* */ (block comment)	&&		-= (compound subtraction)	
// (single line comment)			/= (compound division)	
;			^= (compound bitwise xor)	d
{ } (curly braces)			= (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.

