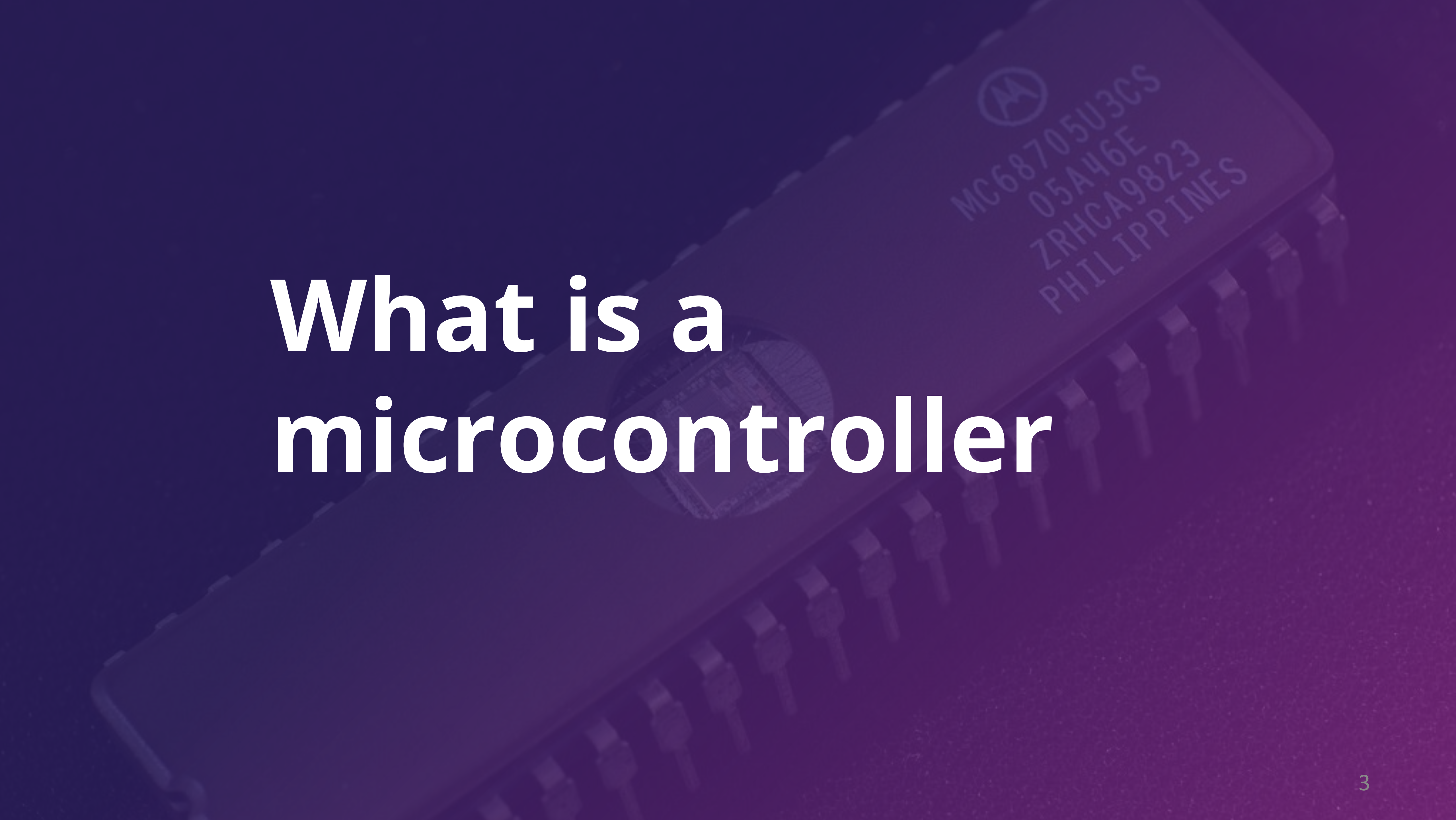
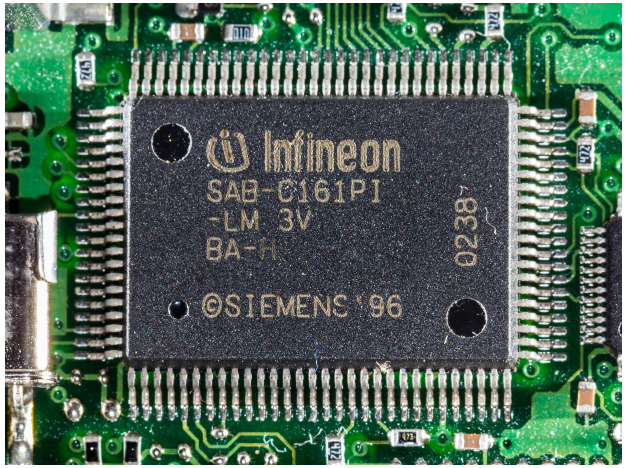
What is a microcontroller



What is a microcontroller?

A microcontroller (MCU for microcontroller unit) is a small computer on a single metal-oxide-semiconductor (MOS) integrated circuit (IC) chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals.



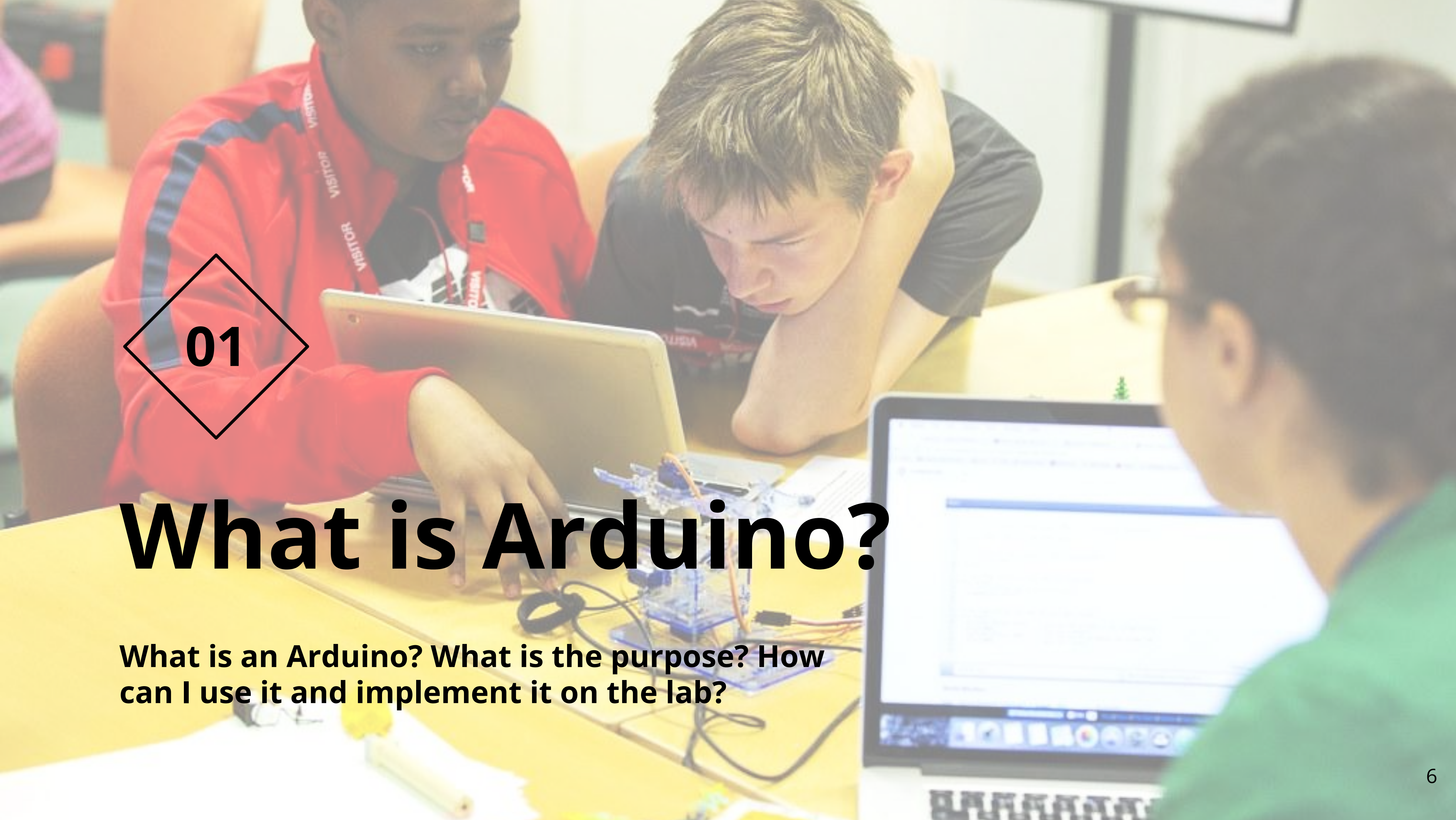
Big computers vs small computers

Dell Precision T1500

- **CPU Speed:** 2.93GHz
  - Quad-core!
- **RAM:** 16GB
- **Storage:** 2TB
- **I/O:**
  - USB, Firewire, Serial, PS/2, RJ-45, Audio, etc.

Atmel ATMEGA328P

- **CPU Speed:** 20MHz
  - Not Quad-core ☺
- **RAM:** 2KB
- **Storage:**
  - 32KB Program Memory
  - 1KB EEPROM
- **I/O:**
  - Up to 23 generic I/O
    - 6 of them ‘analog-capable’
    - UART/SPI/I^{2}C, etc.



01

What is Arduino?

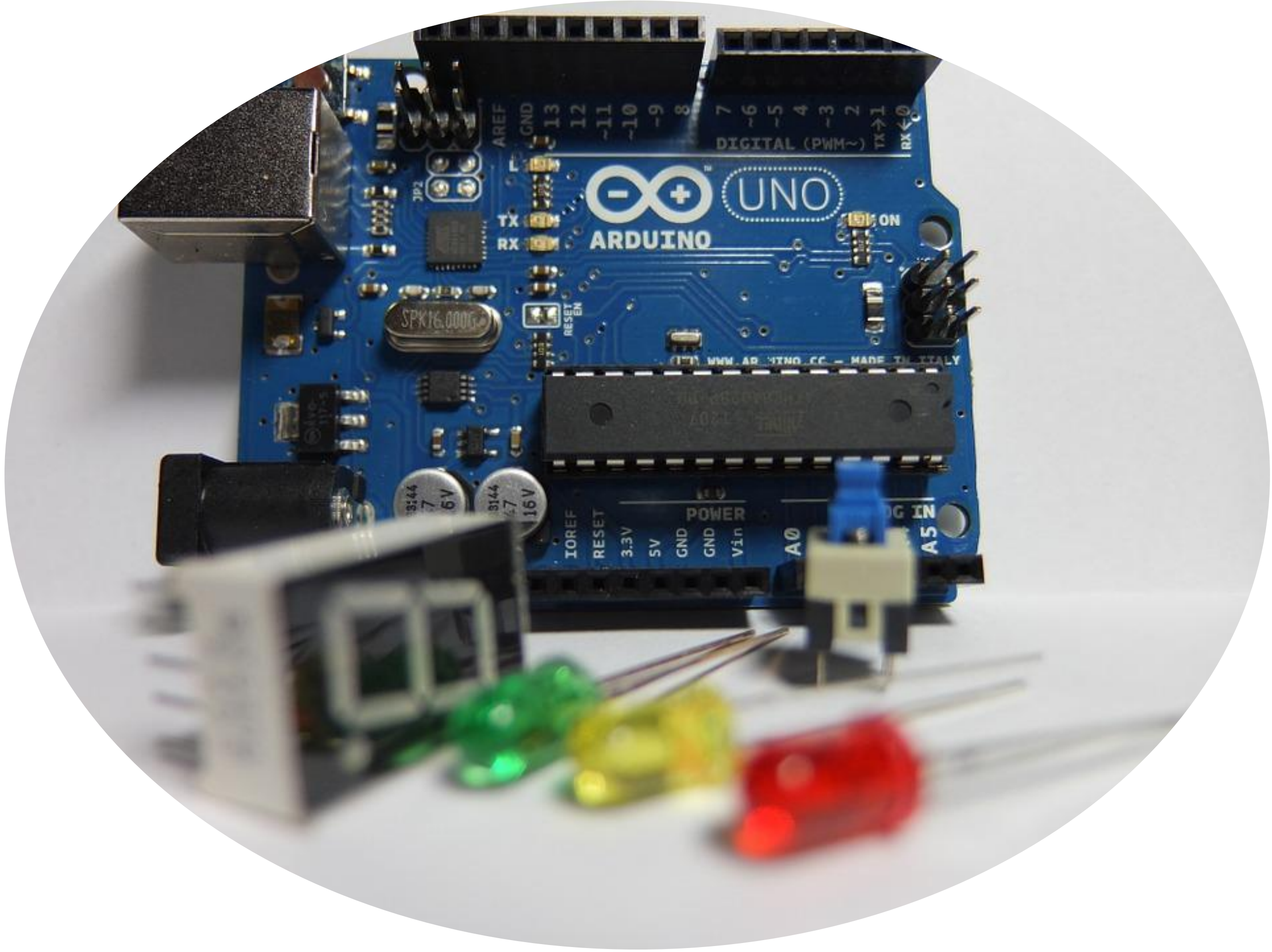
What is an Arduino? What is the purpose? How can I use it and implement it on the lab?

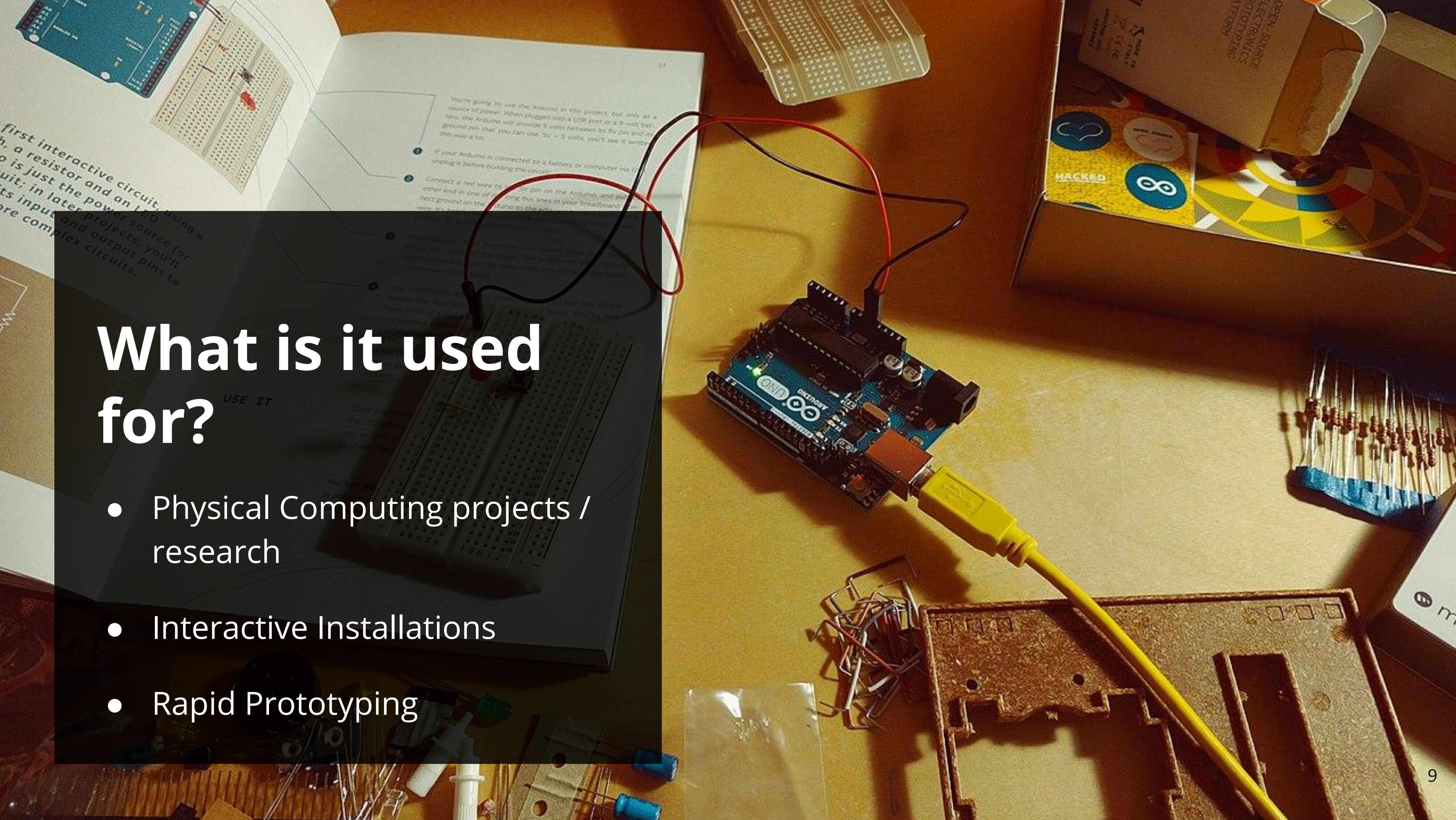
Arduino microcontroller

A brief story. The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators.

Arduino Microcontroller

- Open-source electronics platform based on easy-to-use hardware and software.
- Are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.





What is it used for?

- Physical Computing projects / research
- Interactive Installations
- Rapid Prototyping

What can I do?

Sensors

- Push buttons, touchpads, tilt switches
- Variable resistors (Sliders, Volume knobs)
- Photoresistors (sensing light)
- Thermistors (temperature)
- Ultrasound (proximity range finder)

Actuators

- Lights, LED’s
- Motors
- Speakers
- Displays (LCD’s)

How Arduino is programmed?

Using a software called Arduino IDE

```
  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
}
```

02

Arduino Software

How Arduinos are programmed

Arduino Software (IDE)

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

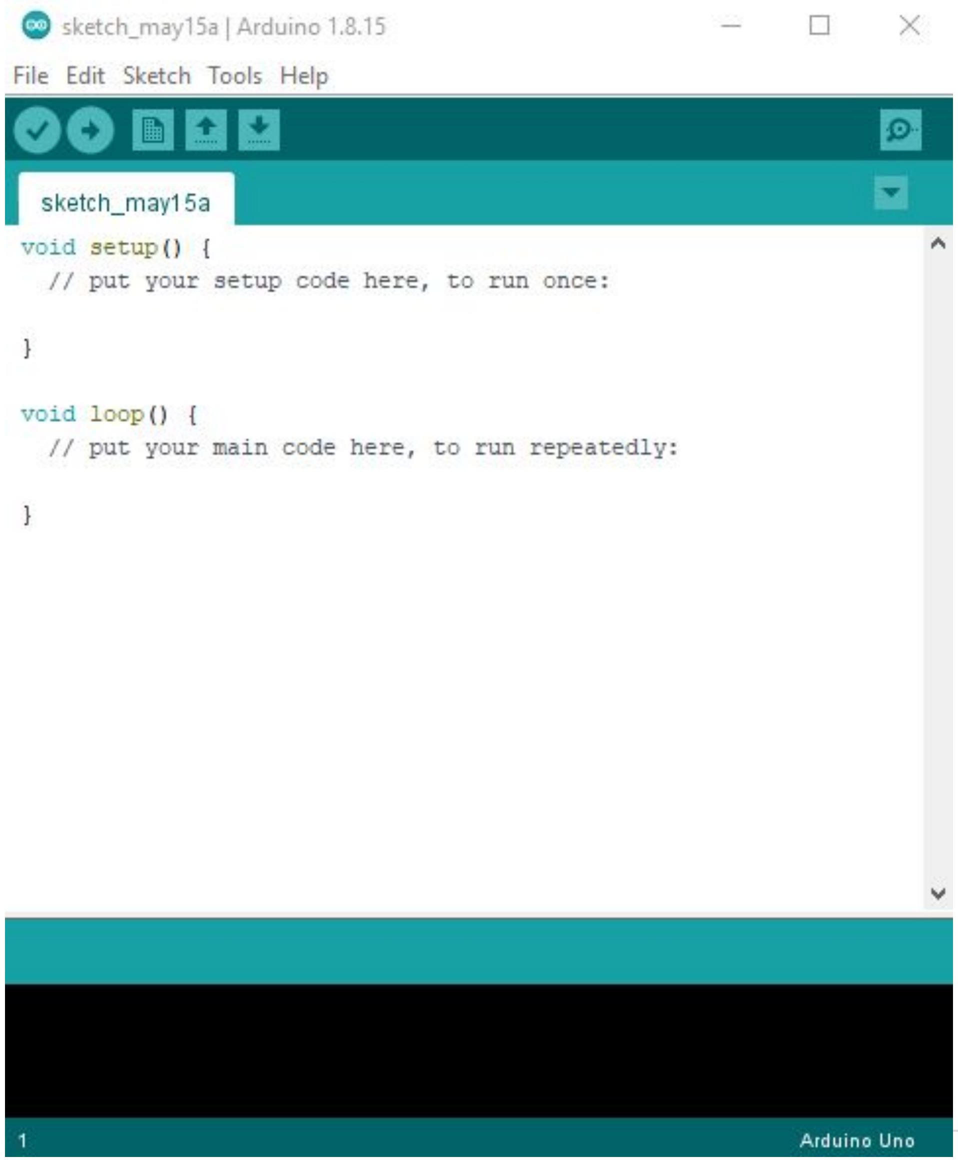
Arduino Language

- Simplified C/C++
- Based on the wiring project
  - http://wiring.org.co
- Peripheral libraries
  - LCD, sensors, 12C, ect.

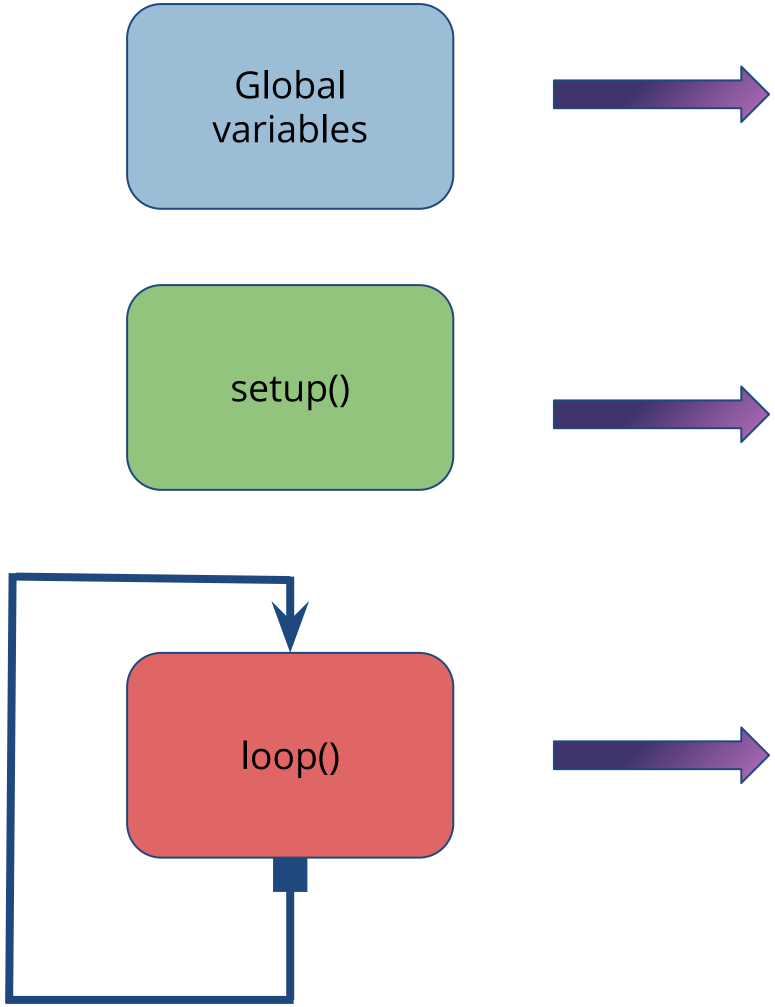
Useful functions

pinMode()	set pin as input or output
digitalWrite()	set a digital pin high/low
digitalRead()	read a digital pin’s state
analogRead()	read an analog pin
analogWrite()	write an “analog” PWM value
delay()	wait an amount of time
millis()	get the current time

Sketch



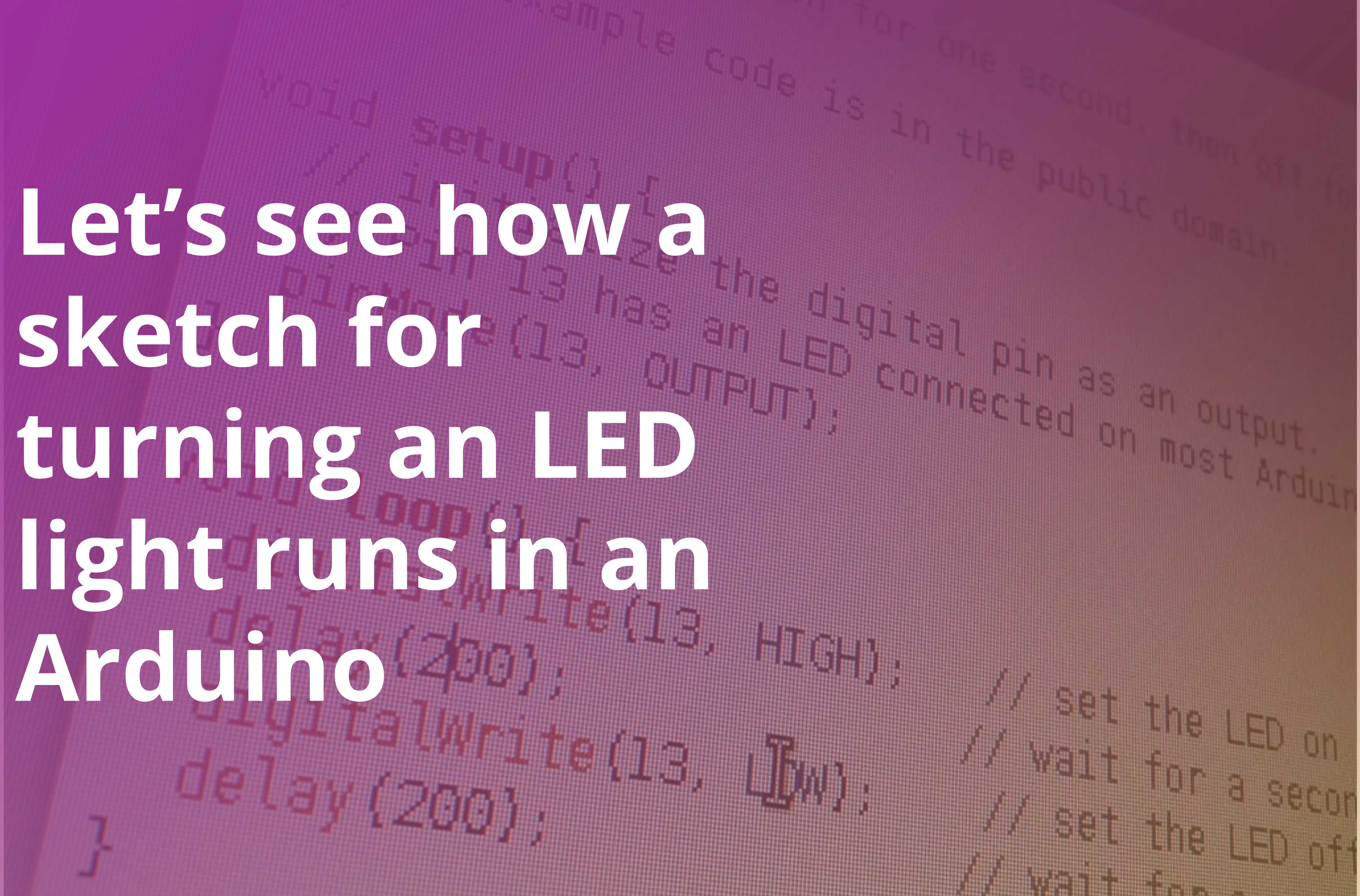
```
void setup() {
  // put your setup code here, to run once:

}

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

}
```



Declare variables at the top

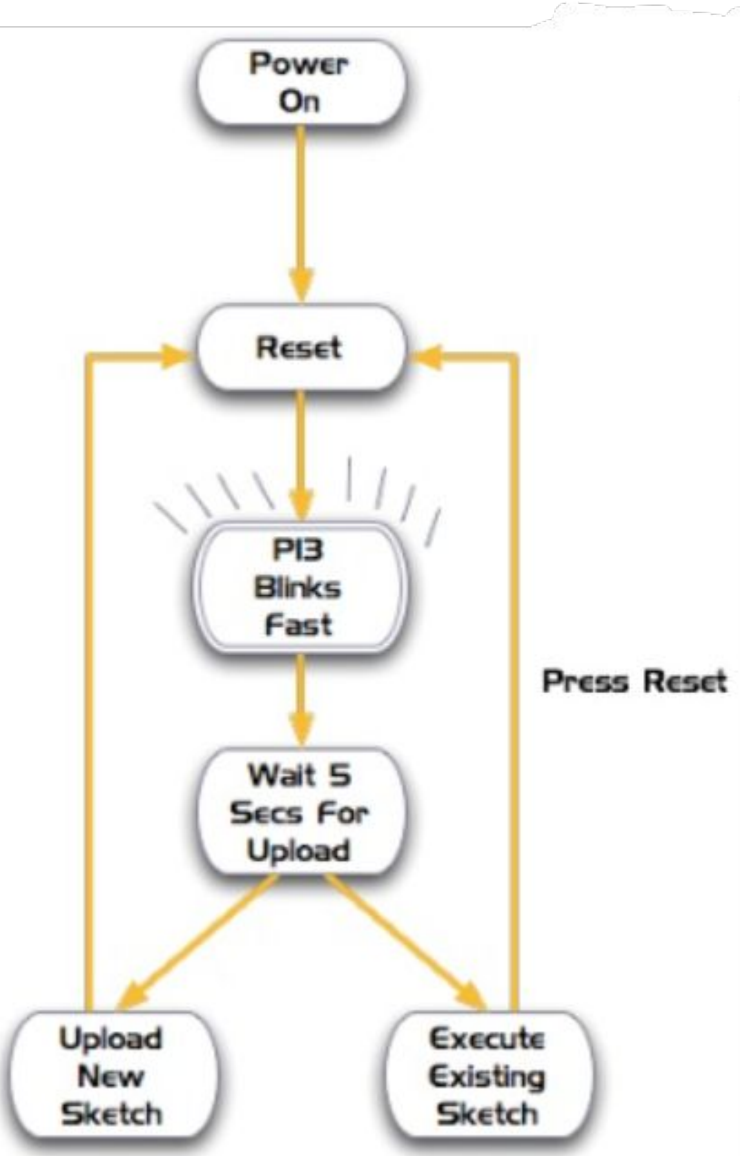
Initialize
setup() - run once at the beginning
set pins

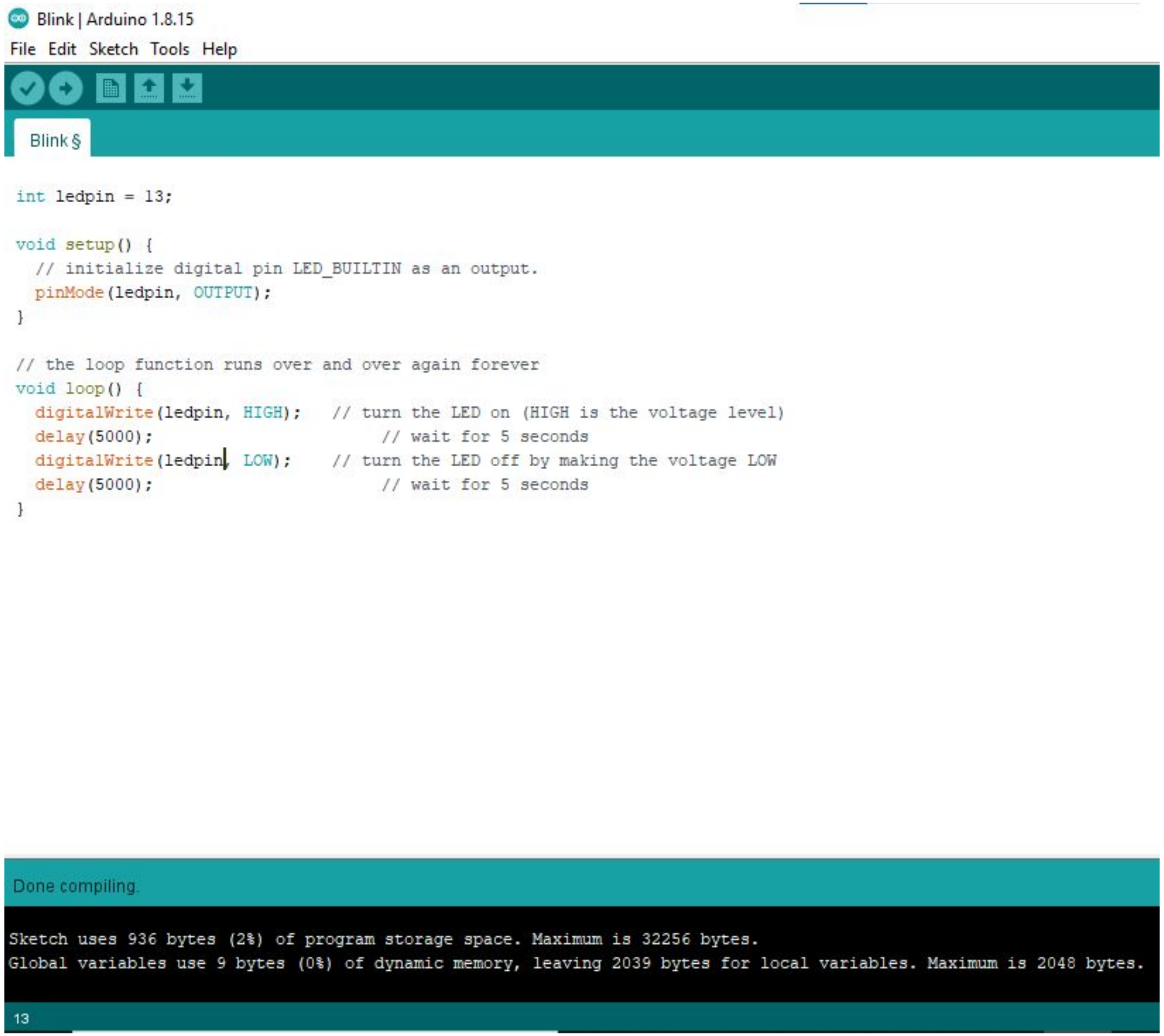
loop() - run repeatedly after setup()



Let’s see how a sketch for turning an LED light runs in an Arduino

Blinking LED



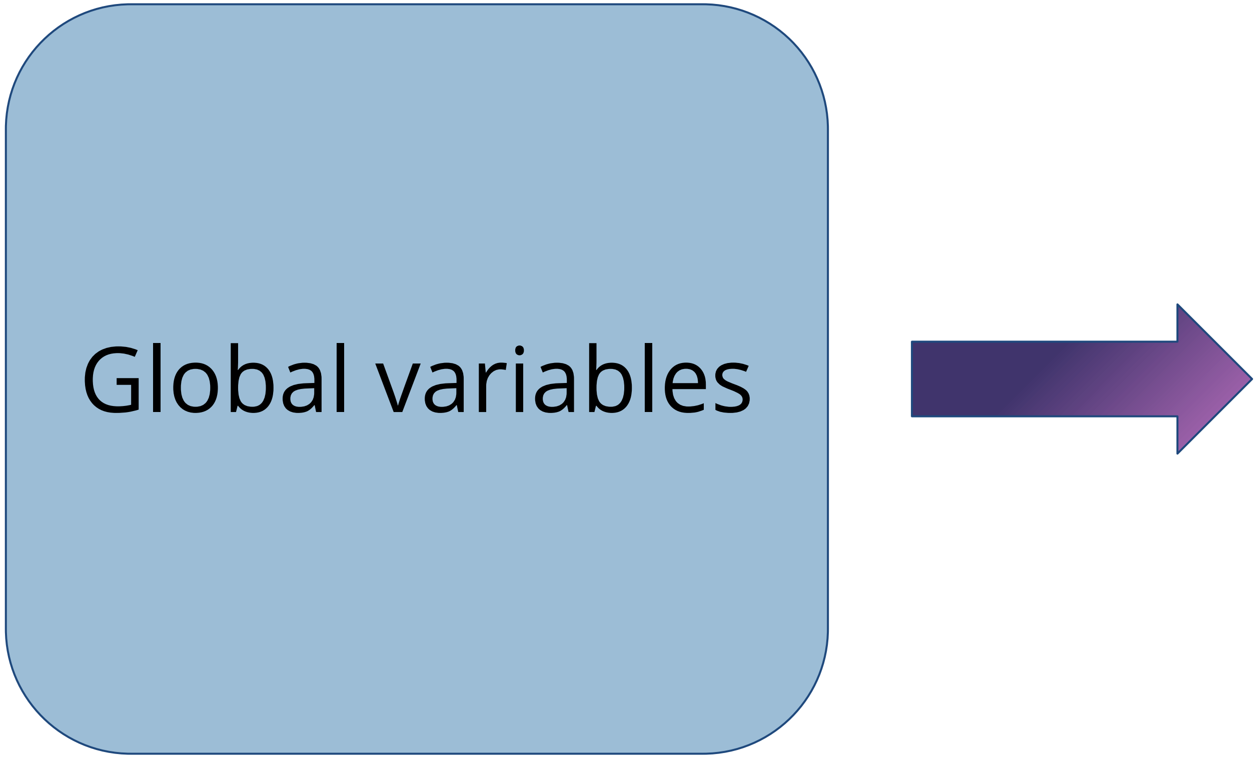


```
int ledpin = 13;

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(ledpin, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledpin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(5000);                      // wait for 5 seconds
  digitalWrite(ledpin, LOW);    // turn the LED off by making the voltage LOW
  delay(5000);                      // wait for 5 seconds
}
```

Done compiling.

```
Sketch uses 936 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

Global variables

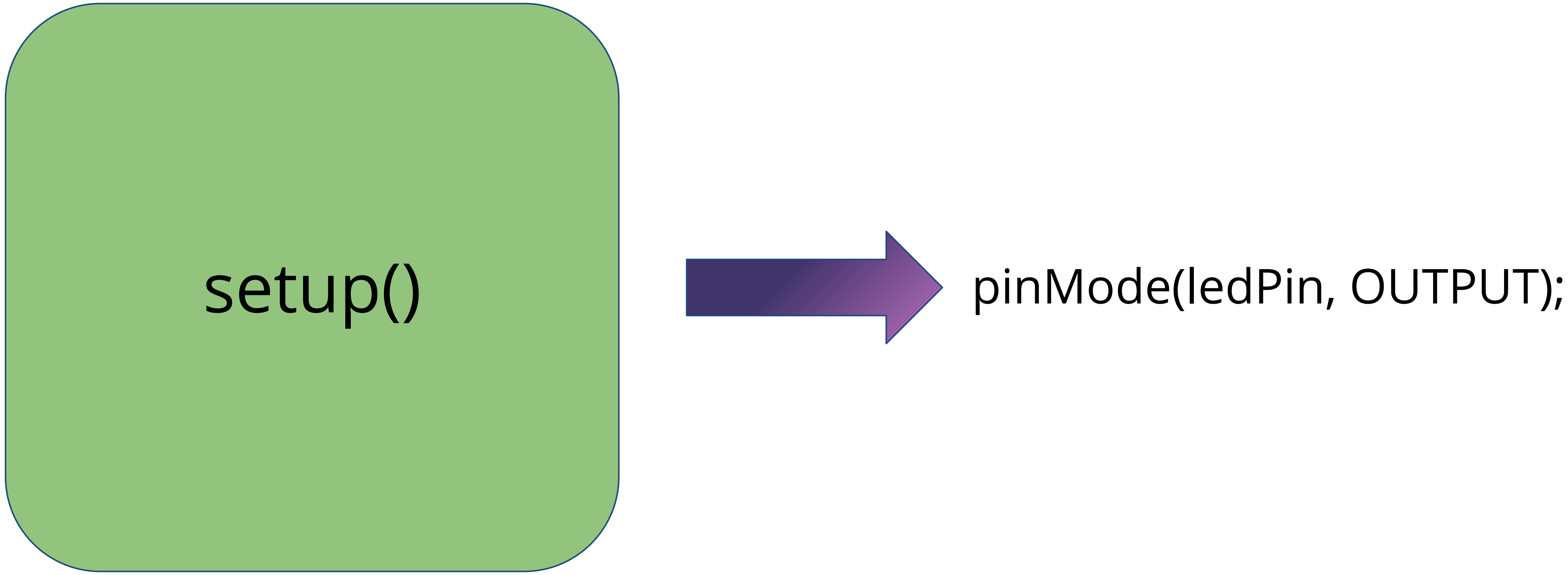


Global variables

int ledPin = 13;

- LED connected to the control pin 13

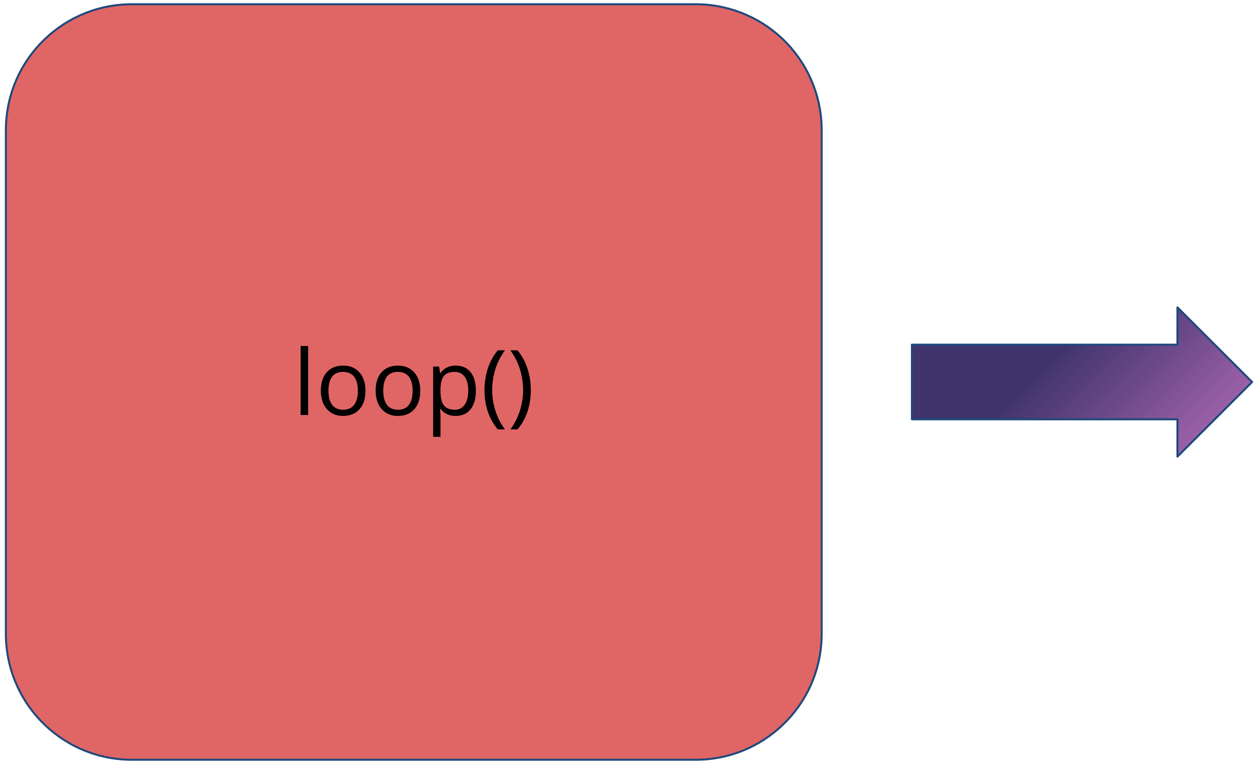
Setup()



setup()

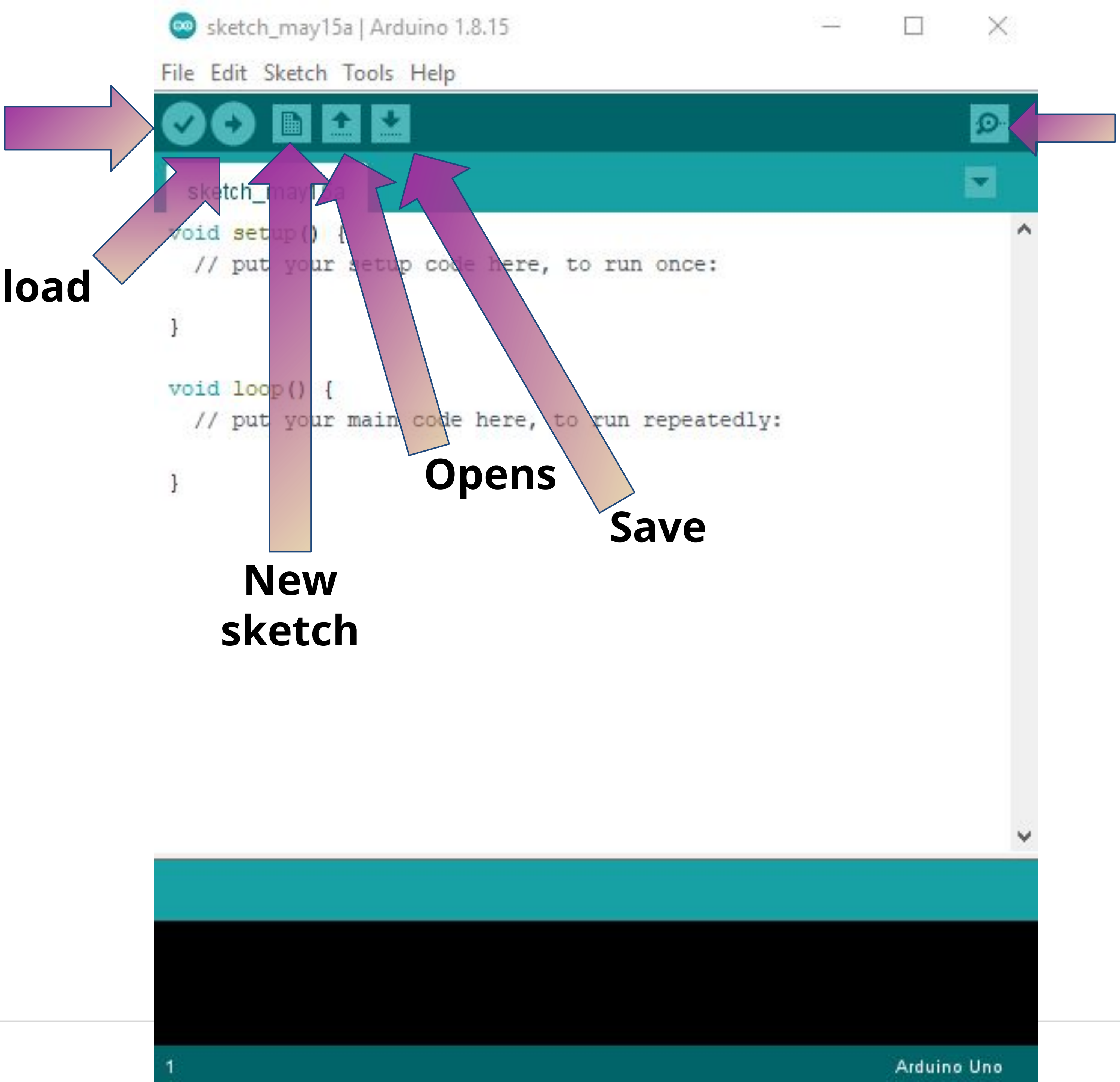
```
pinMode(ledPin, OUTPUT);
```

loop()



loop()

```
digitalWrite(ledPin, HIGH);
delay(5000);
digitalWrite(ledPin, LOW);
delay(5000);
```



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

}

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

}
```

Verify

Upload

New sketch

Opens

Save

Search

More projects

You can find more projects at:

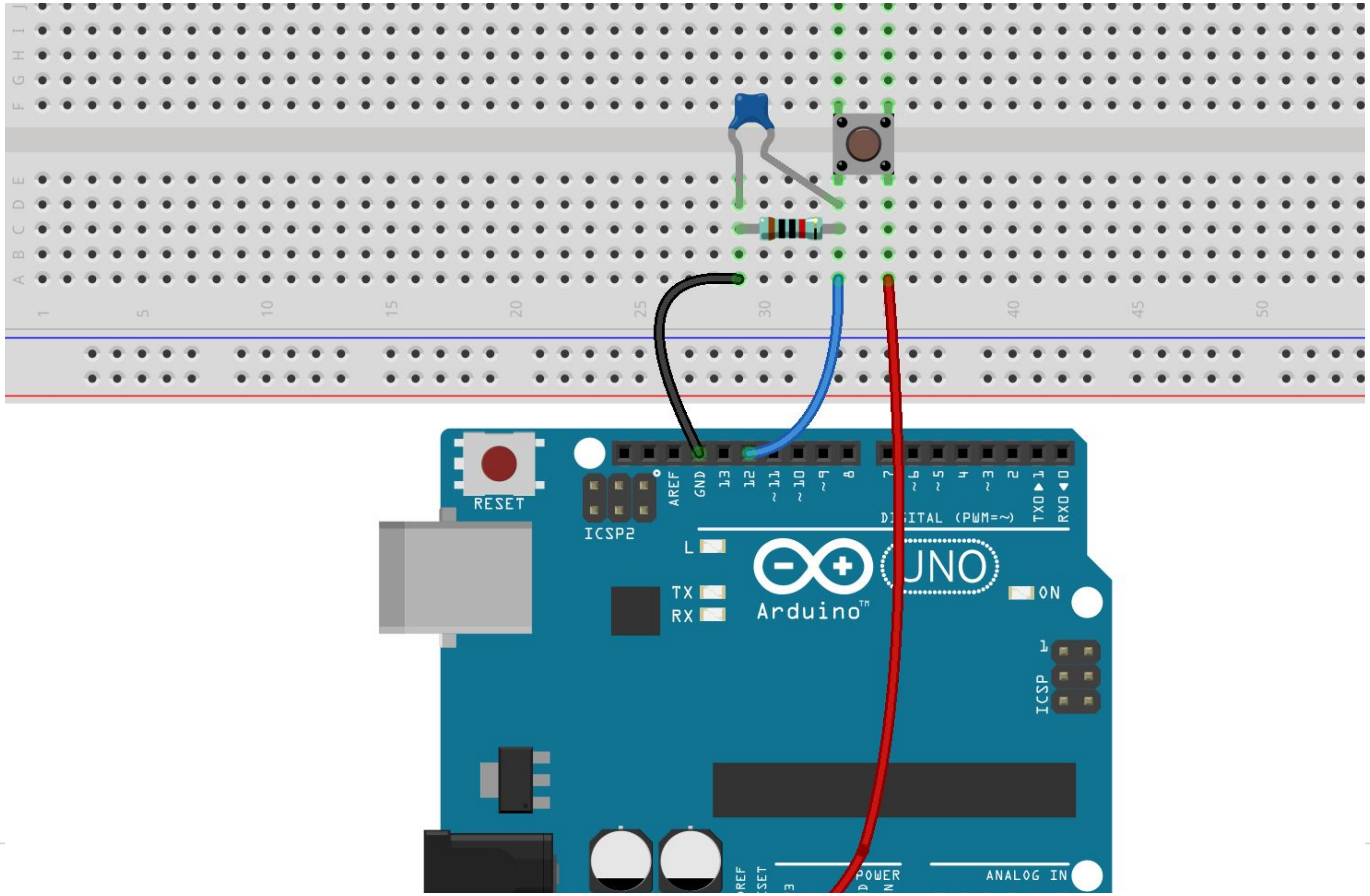
https://create.arduino.cc/projecthub

04

Hands On

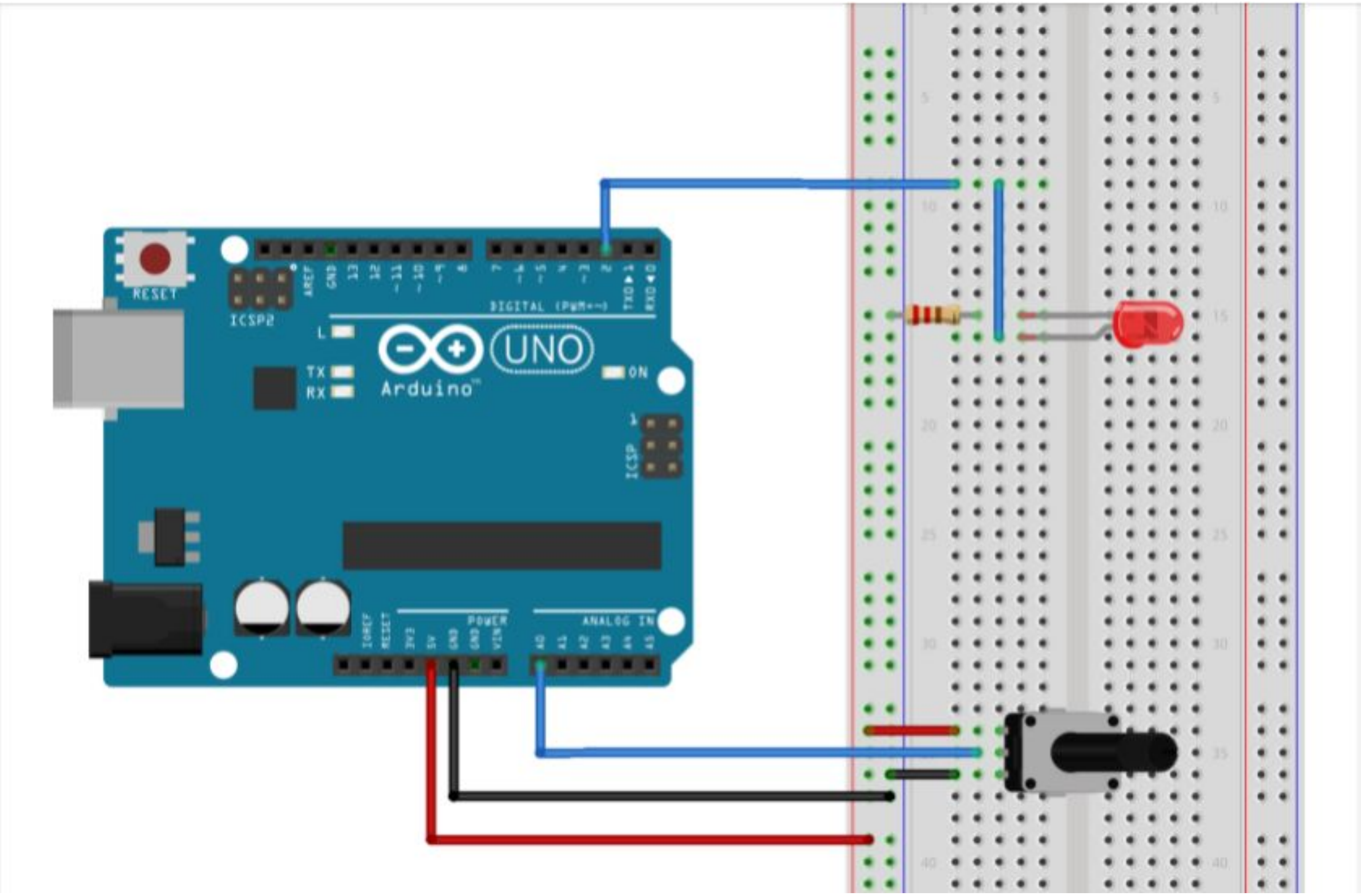
Challenge!!

LED Control Using a button



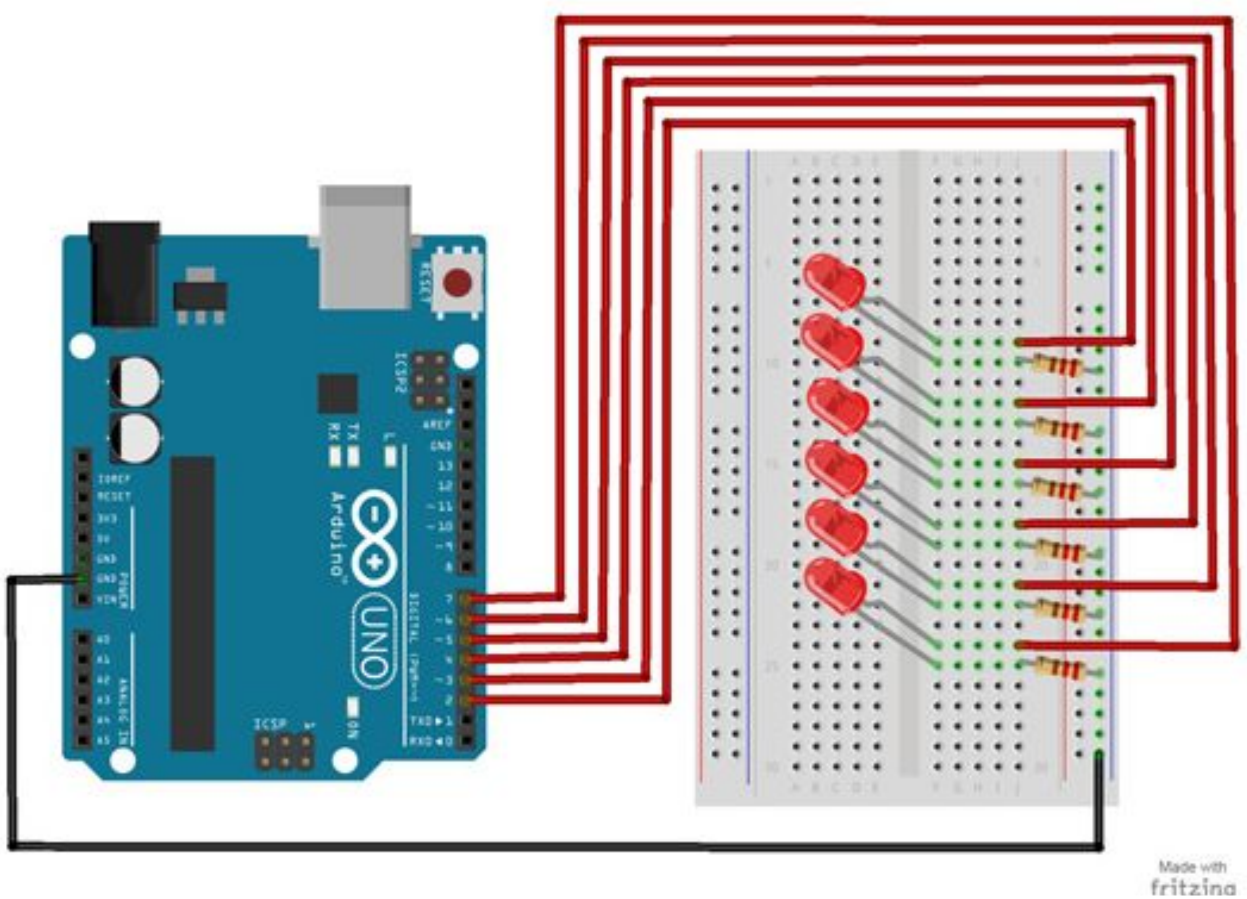
Challenge!!

LED Brightness Control Using a Potentiometer



Challenge!!

Scrolling LED


Arduino
UNO
RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM~)
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
ICSP
ON
TX
RX
L
Made with
fritzing