



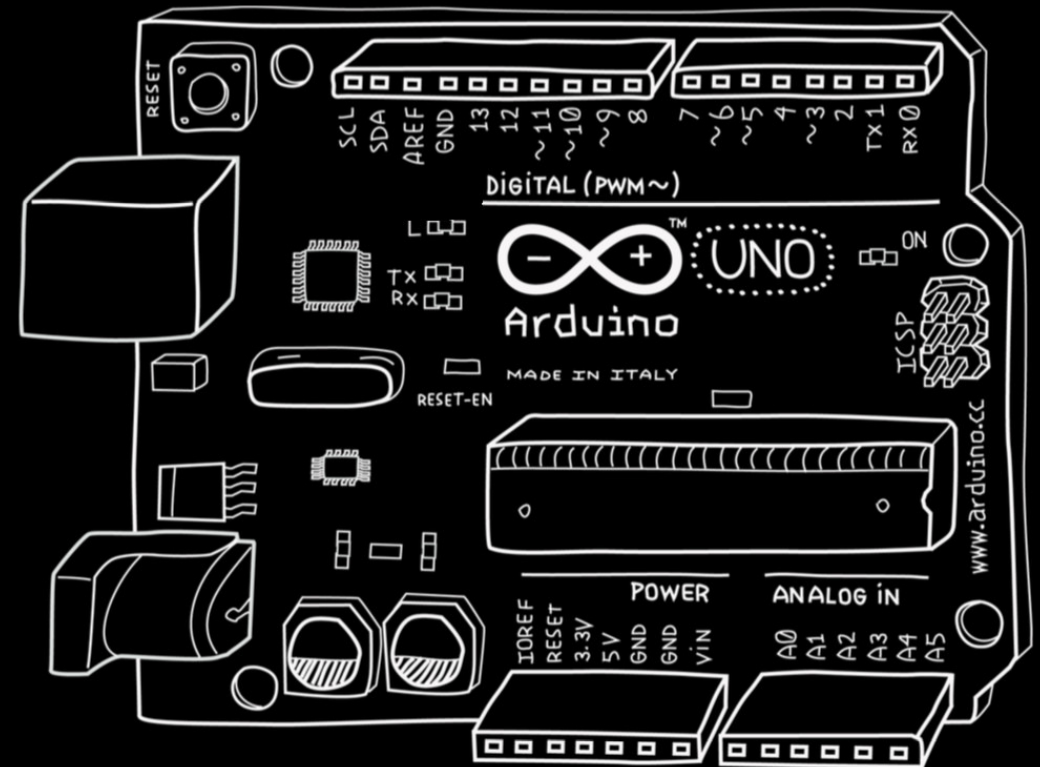
INTRO TO HARDWARE

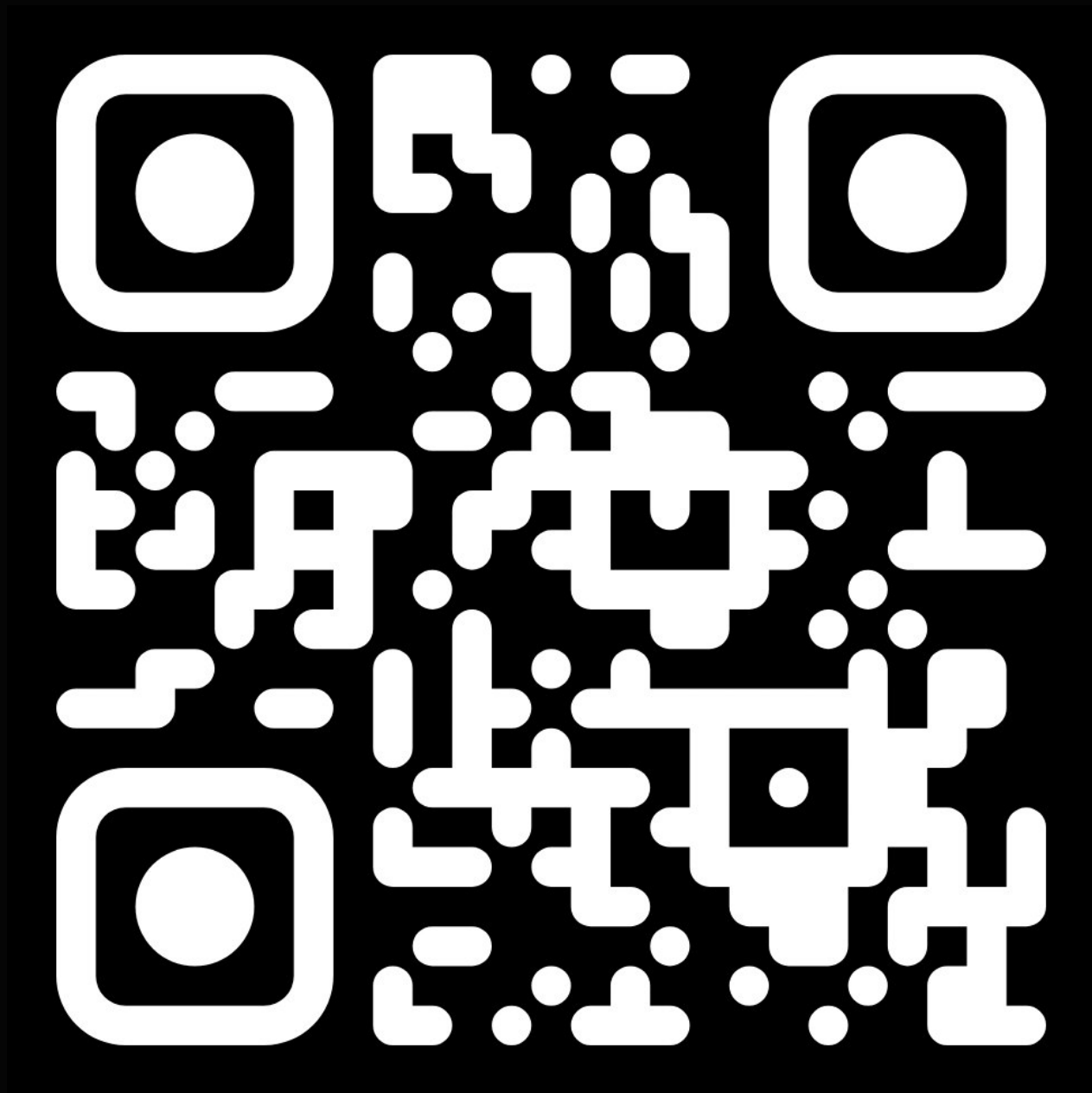
(ARDUINO EDITION)

Basics & I/O

Introduction to Arduino Hardware and Programming

By: Abhijith R





learn.jithurx.dev



TODAY'S AGENDA

// Digital Output

Blink, Traffic Light.

// Digital Input

Push Button.

// PWM

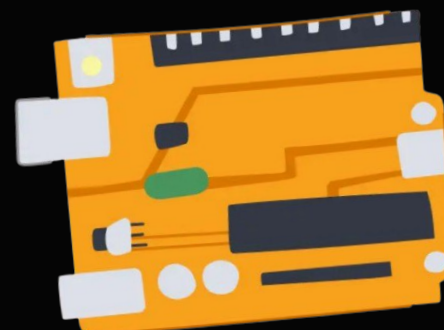
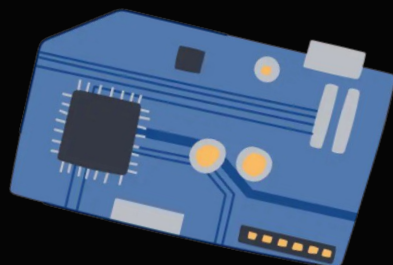
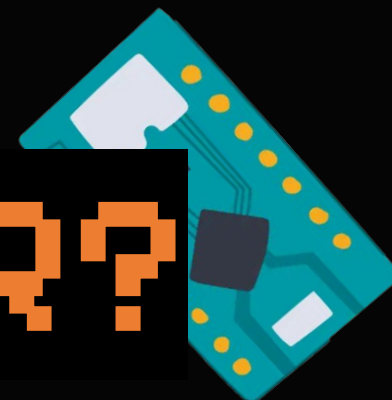
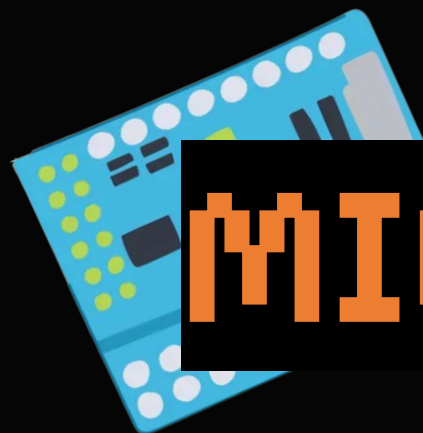
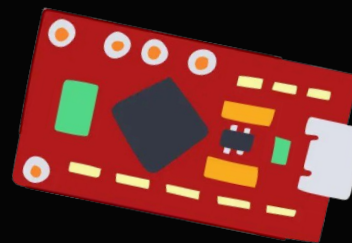
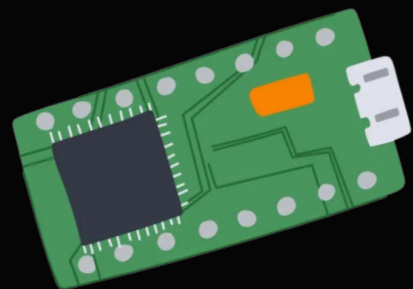
Fade LED.

// Mini Challenge

build something of your own.



WHAT IS A MICROCONTROLLER?





WHAT IS A MICROCONTROLLER?

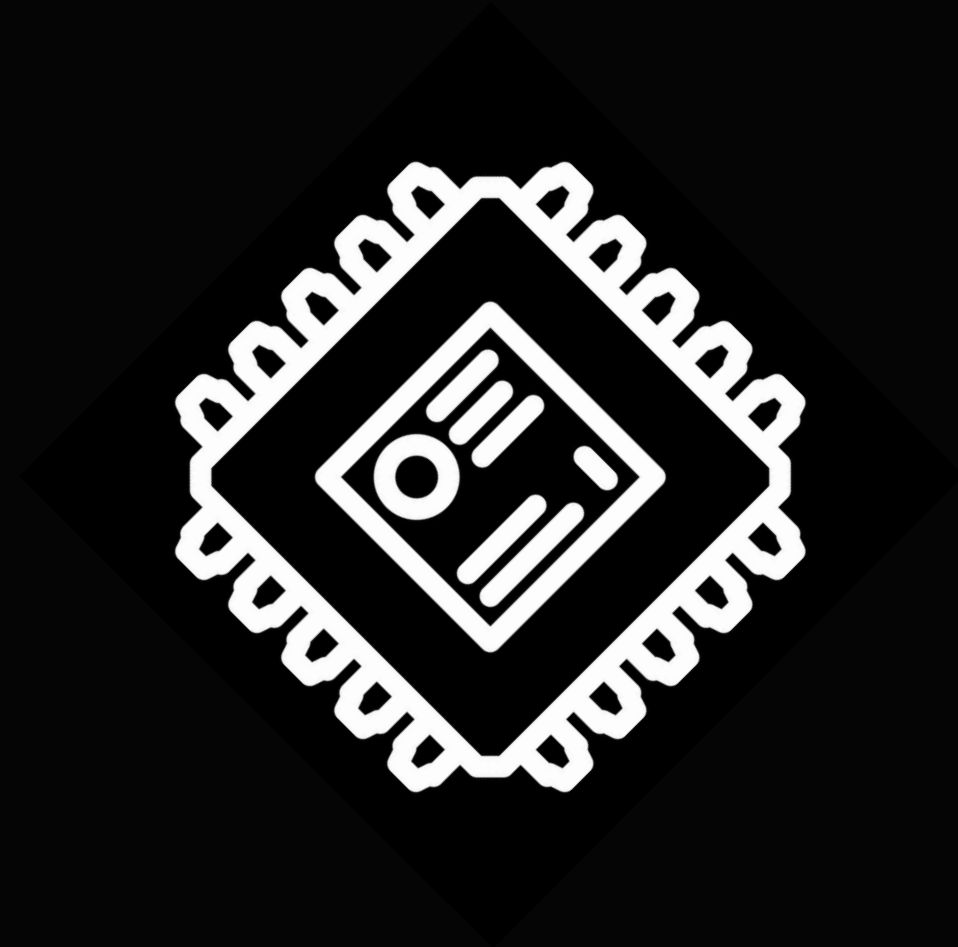
Hardware

A tiny computer on a single chip

CPU + RAM + Flash + I/O pins

Runs one program, repeatedly, forever

Software





THE ARDUINO ECOSYSTEM

About

Arduino is an open-source electronics platform based on easy-to-use hardware and software

Ecosystem

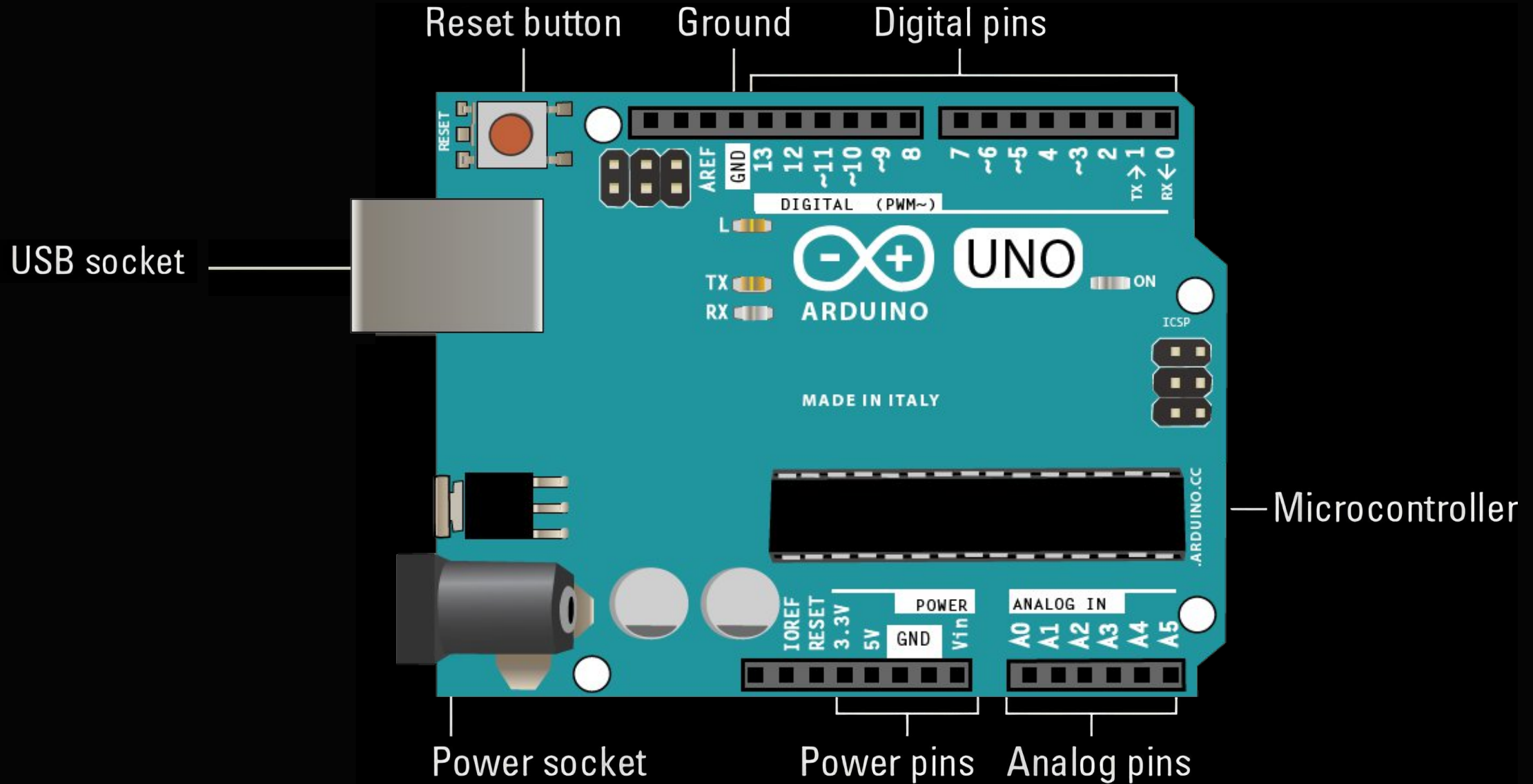
- Micro-controller board
- IDE (software)
- Libraries
- Community

Popularity

- Beginner-Friendly
- Simplified C/C++
- Huge community
- Pre-built libraries



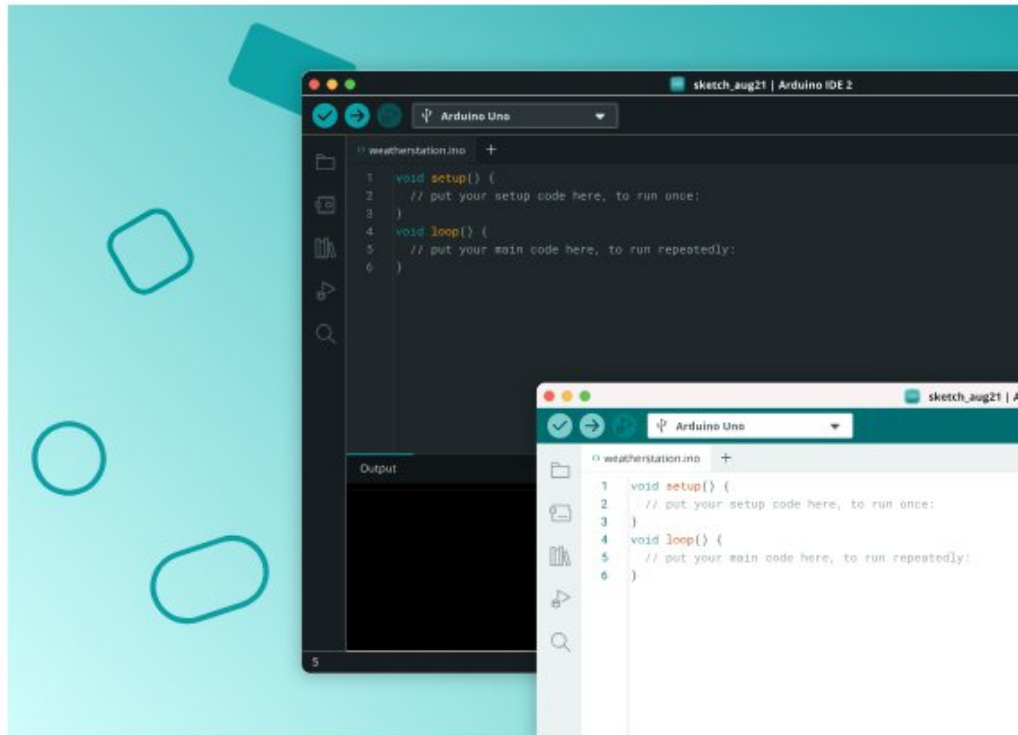
ANATOMY OF THE ARDUINO UNO



INSTALLING THE ARDUINO IDE



Go to : www.arduino.cc/en/software



Arduino IDE 2.3.10

[Release notes](#)

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger. For more details, check out the [Arduino IDE 2.0 documentation](#).

Linux Appliance (64-bit X86-64)

DOWNLOAD



Nightly Builds

Download a preview of the incoming release with the most updated features and bugfixes.

Linux Appliance 64 bits (X86-64)

DOWNLOAD

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).



Arduino IDE 2.3.10

Release notes

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger. For more details, check the [Arduino IDE 2.0 documentation](#).

Linux AppImage (64-bit X86-64)

DOWNLOAD

Windows Win 10 or newer (64-bit)

Windows MSI installer

Windows ZIP file

Linux AppImage (64-bit X86-64)

Linux ZIP file (64-bit X86-64)

macOS Intel 12 Monterey or newer (64-bit)

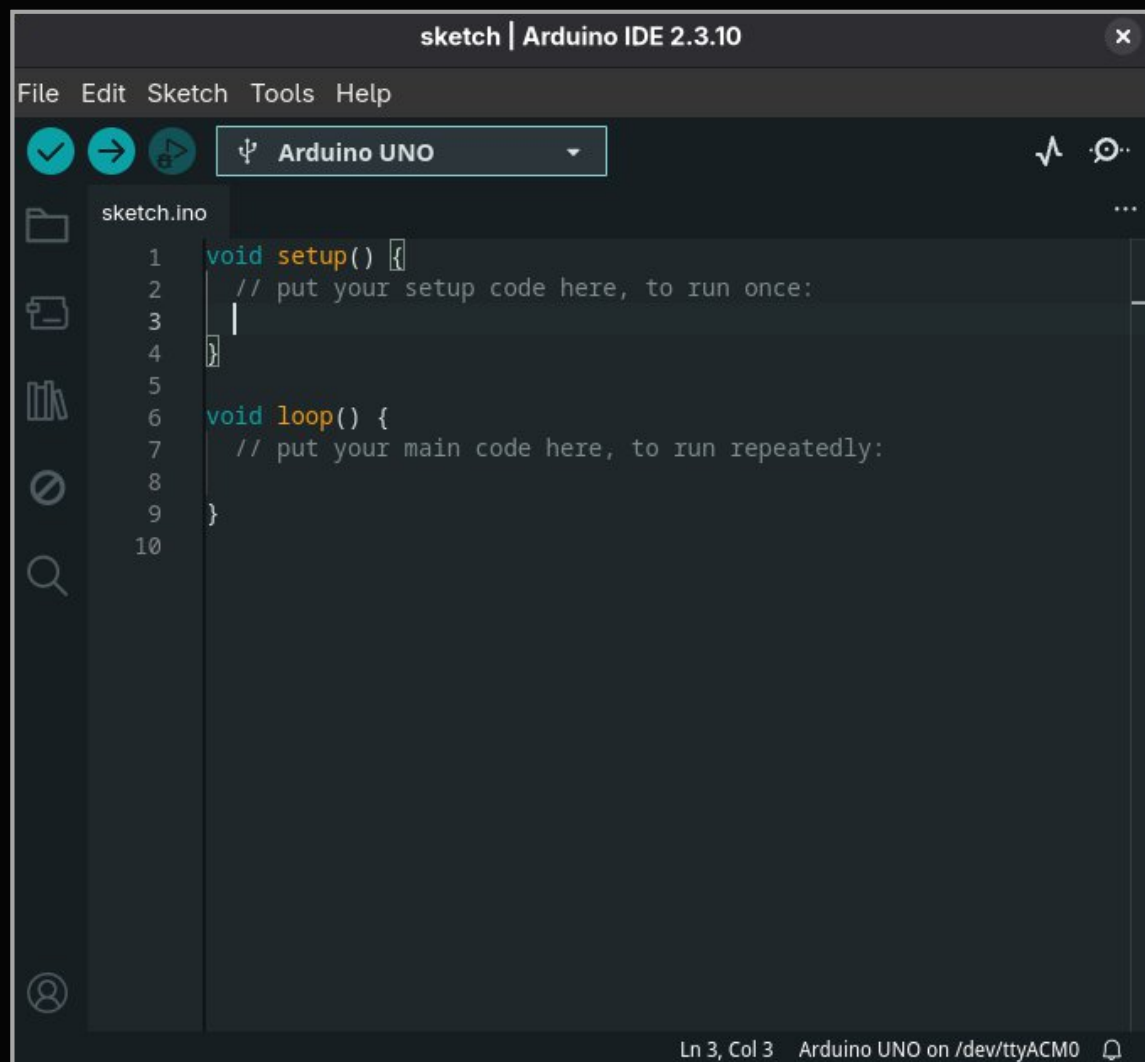
macOS Apple Silicon 12 Monterey or newer (64-bit)

Select Installer for your OS

Download and Install



IDE layout



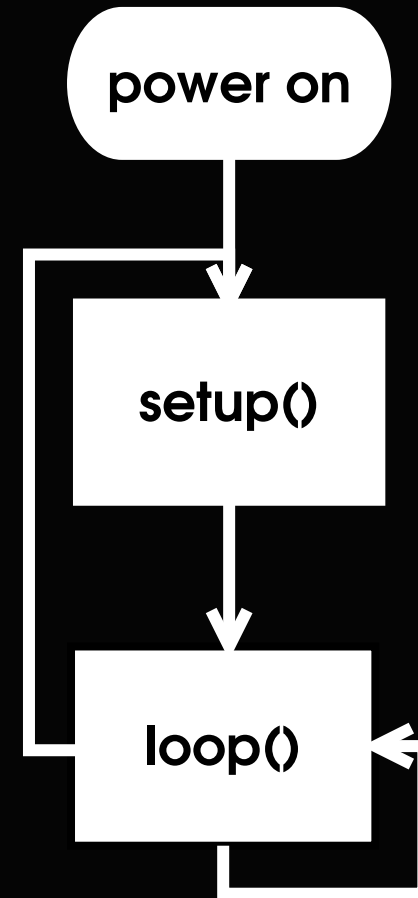


ANATOMY OF A SKETCH

sketch | Arduino IDE

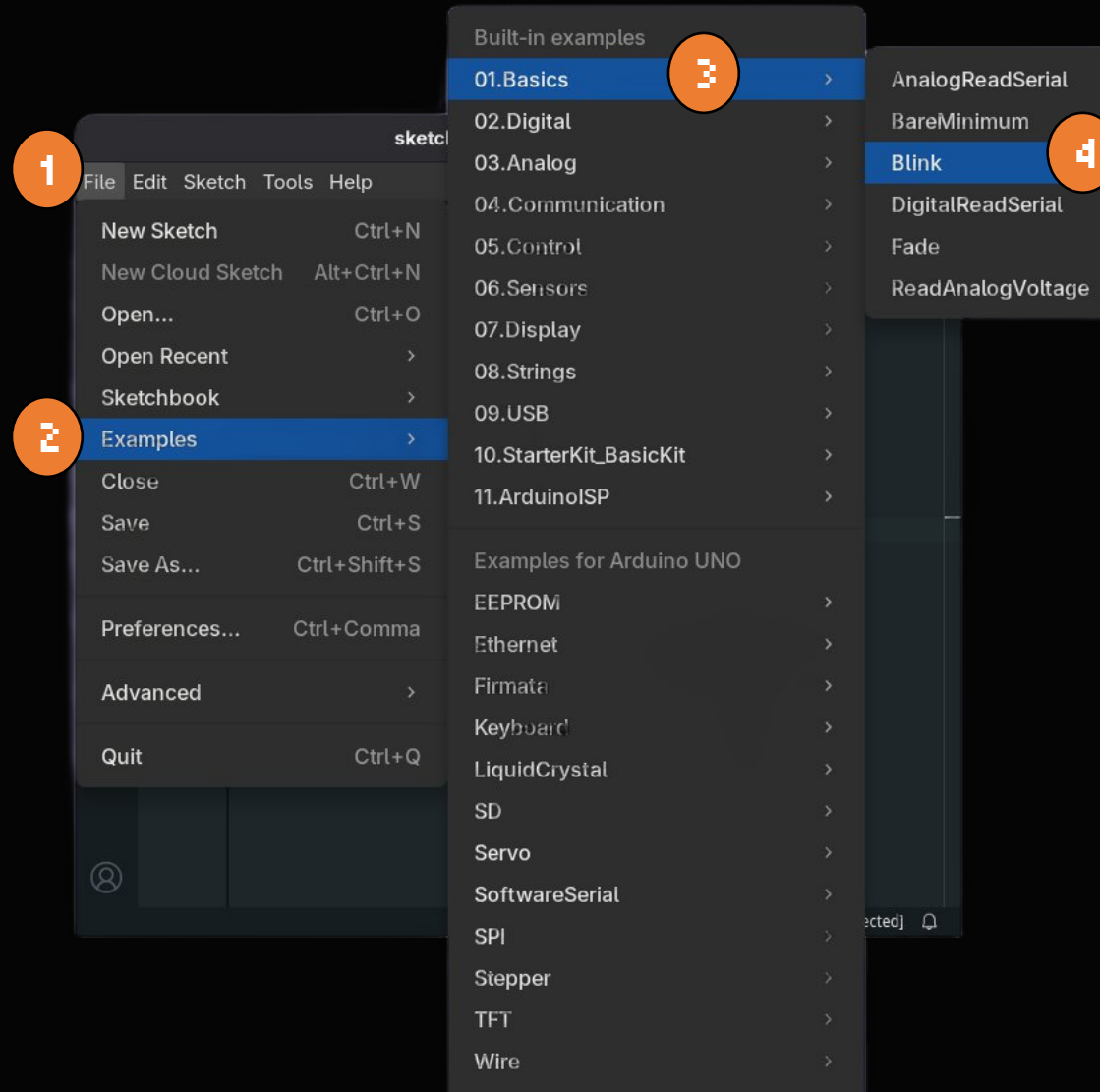
```
void setup() {           // runs once
  // put your setup code here:
}

void loop() {            // runs forever
  // put your main code here:
}
```





UPLOADING YOUR FIRST SKETCH





Blink | Arduino IDE 2.3.10

File **5** Sketch Tools Help

✓ → ⚙️ Arduino UNO

Blink.ino

```
23 //
24
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27   // initialize digital pin LED_BUILTIN as an output.
28   pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33   digitalWrite(LED_BUILTIN, HIGH); // change state of the LED by setting the pin to the HIGH voltage level
34   delay(1000); // wait for a second
35   digitalWrite(LED_BUILTIN, LOW); // change state of the LED by setting the pin to the LOW voltage level
36   delay(1000); // wait for a second
37 }
38
```

Output

Sketch uses 924 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.

Ln 1, Col 1 Arduino UNO on /dev/ttyACM0 2



WHAT IS DIGITAL OUTPUT?

Digital Pin States

HIGH

(5V)

OR

LOW

(0V)

digital = only two states.

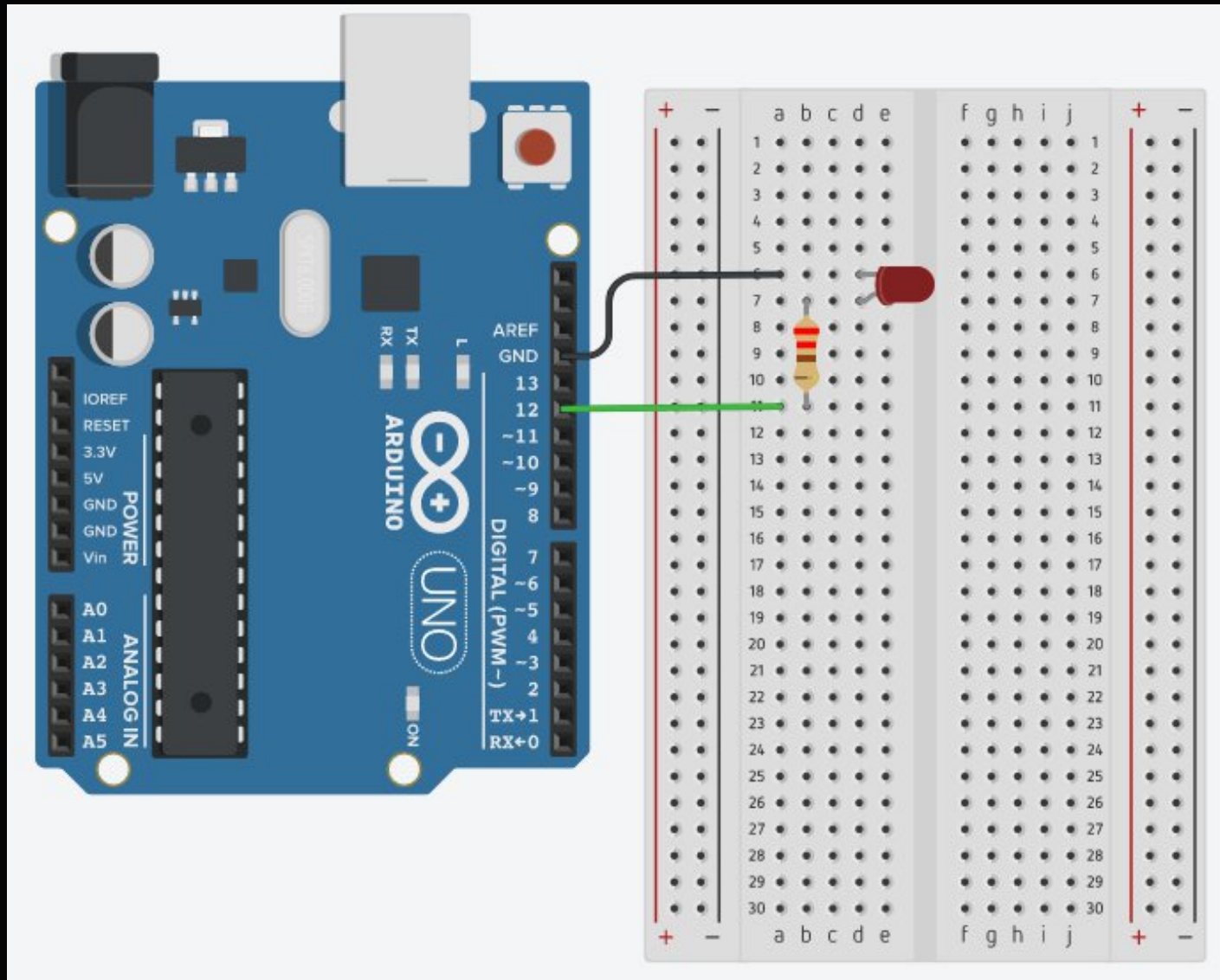
About

Act as a switch controlled by code

Attachments

- LEDs
- Relays
- Buzzers
- Motors (via driver)
- etc

CIRCUIT: WIRING AN LED



DON'T FORGET RESISTOR

LEDs have very low internal resistance

220 Ω limits current to a safe **$\sim 15-20$ m**



Function: pinMode

sketch.ino

```
pinMode(8, OUTPUT);
```

- Tells the Arduino: "this pin will send signals out"
- Must be set inside setup()

Other mode: INPUT



Function: digitalWrite

sketch.ino

```
digitalWrite(8, HIGH); // turn ON  
digitalWrite(8, LOW);  // turn OFF
```

- Sets a digital pin's voltage

HIGH = 5V,

LOW = 0V



Function: delay()

sketch.ino

```
delay(1000); // pause for 1000 milliseconds = 1 second
```

- Pauses the program for a given time, in milliseconds
- Nothing else happens on the board while delay() is running



Full Blink Code Walkthrough

sketch.ino

```
int ledPin = 8;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```



Build & Upload Blink



TASK

Wire the LED circuit and write / upload the code from the last slide.

STRETCH GOALS

- Make it blink twice as fast
- Blink an SOS-like pattern manually (no loop)





Variables in Arduino

sketch.ino

```
int ledPin = 8;    // whole numbers  
bool isOn = true;  // true/false  
long delayTime = 500; // larger numbers
```

- A variable is a named "box" that stores a value
- Using variables instead of raw numbers makes code easier to read and change

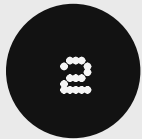


Timing Patterns with delay

- Different ON/OFF durations create different "feels": fast = urgent, slow = calm
- Sequencing multiple digitalWrite + delay pairs creates a pattern
- A traffic light is exactly that: a fixed sequence of timed states



Traffic Light System



TASK

Build a 3-LED traffic light: Red → Green → Yellow → Red...

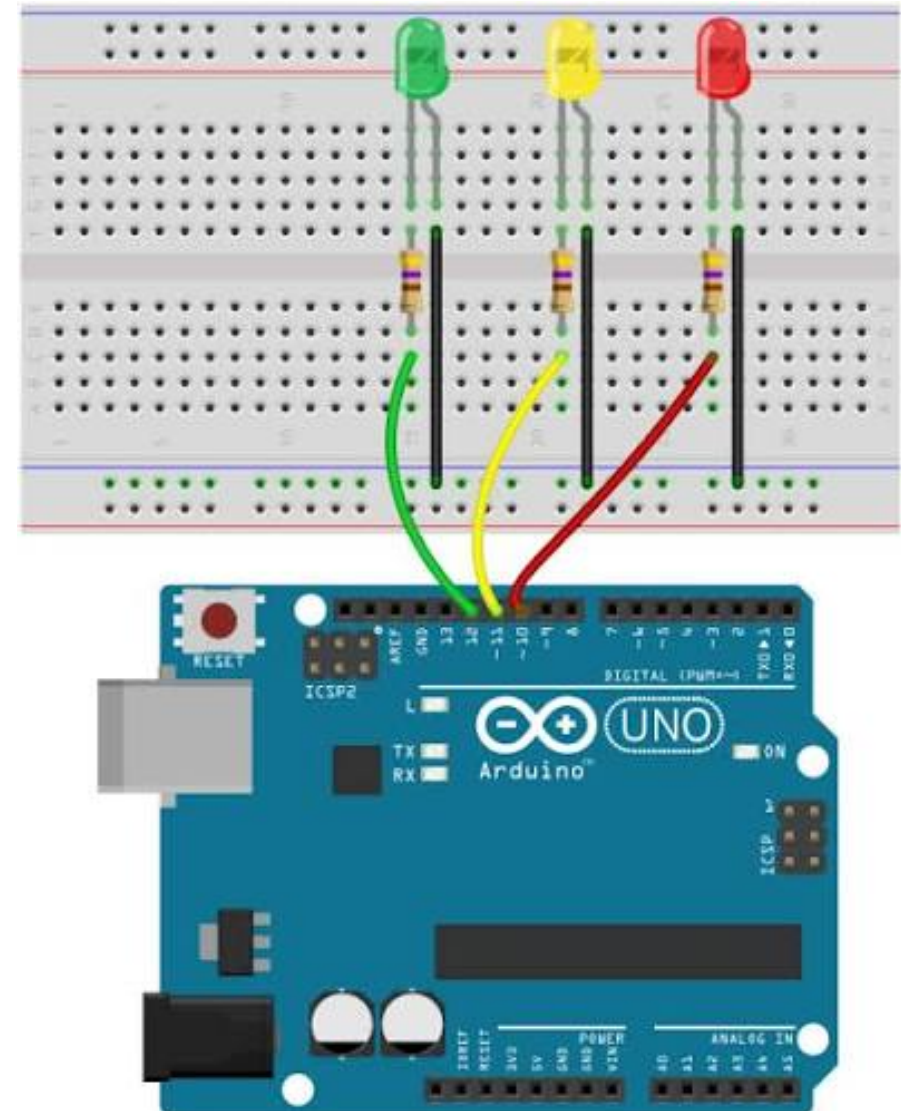
STRETCH GOALS

- Suggested timing: Red 3s, Green 3s, Yellow 1s
- Use 3 separate digital pins, each LED with its own 220 Ω resistor



Traffic Light - Circuit Diagram

- Red LED → pin 8 (+ resistor) → GND
- Yellow LED → pin 9 (+ resistor) → GND
- Green LED → pin 10 (+ resistor) → GND





Traffic Light - Code Hints

sketch.ino

```
int red = 8, yellow = 9, green = 10;

void setup() {
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}

void loop() {
  // your sequence here:
  // red on -> wait -> red off, green on -> wait -> ...
}
```

- Think about what needs to turn OFF before the next color turns ON



Build the Traffic Light



TASK

Complete the wiring and loop() logic for the full Red → Green → Yellow cycle.

STRETCH GOALS

- Add a "blinking yellow" caution mode before switching to red

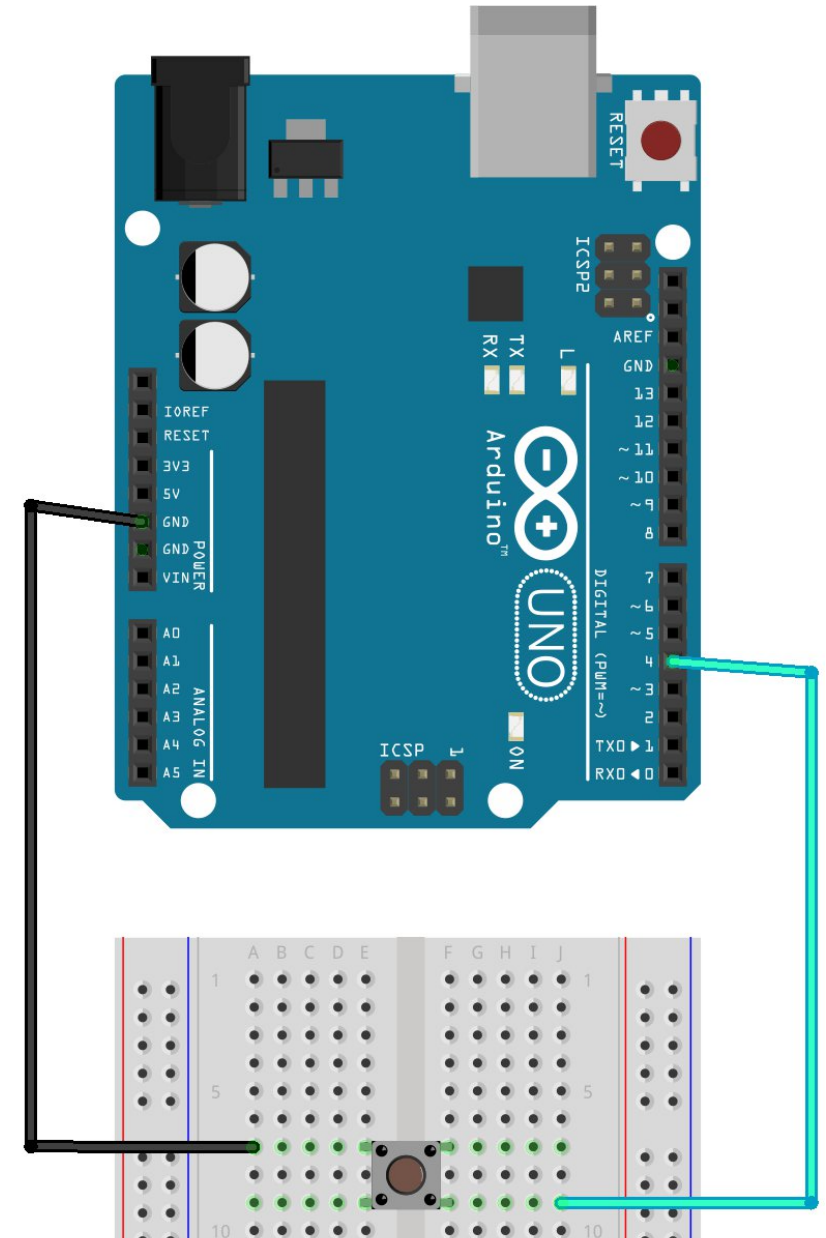


What Is Digital Input?

- Instead of sending a signal, the Arduino now reads one
- A digital input pin reports HIGH or LOW based on an external signal — e.g. a button press
- Real-world examples: door sensors, keypad buttons, limit switches

Push Button Circuit

- Button across two breadboard rows
- One leg → Arduino digital pin (e.g. pin 2)
- Other leg → GND
- No external resistor needed (internal pull-up — next slide)





The Internal Pull-Up Resistor

sketch.ino

```
pinMode(2, INPUT_PULLUP);
```

- Problem: an unconnected ("floating") input pin reads random noise
- Solution: INPUT_PULLUP connects an internal resistor to 5V by default
- Result: HIGH when not pressed, LOW when pressed — inverted logic!



Function: digitalRead

sketch.ino

```
int state = digitalRead(2);  
if (state == LOW) {  
    // button is being pressed  
}
```

- Reads the current HIGH/LOW value of a digital pin
- Store the result in a variable to use in an if statement



Button-Controlled LED - Full Code

sketch.ino

```
int buttonPin = 2;
int ledPin = 8;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int state = digitalRead(buttonPin);
  if (state == LOW) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```



Build the Button-Controlled LED



TASK

Wire the button + LED, then upload the code from the last slide.

STRETCH GOALS

- Make the button toggle the LED — stays on after release, turns off on next press



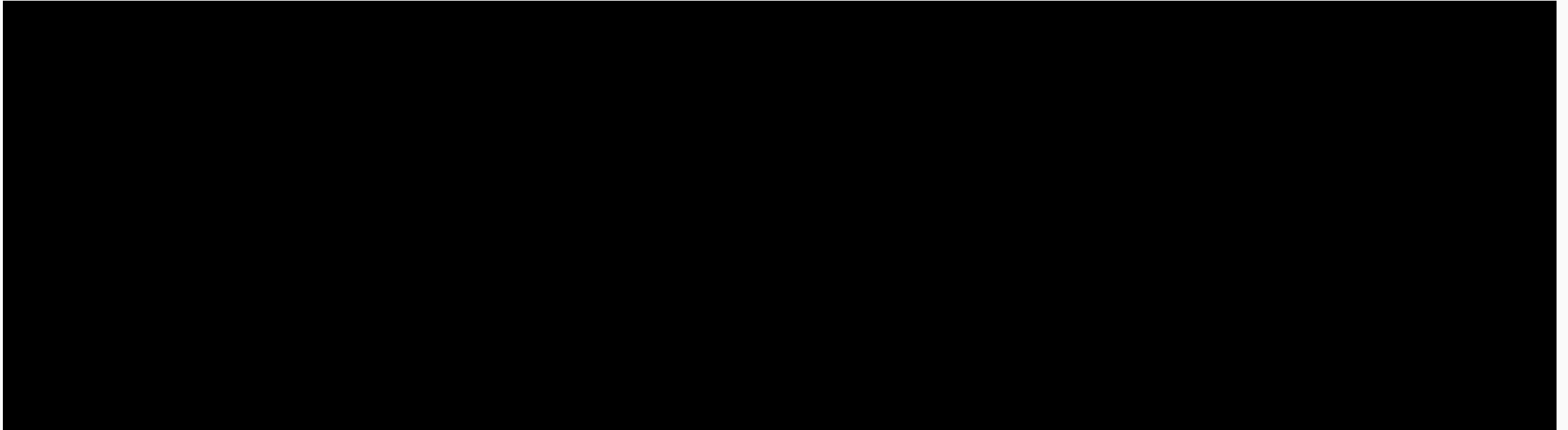
What Is PWM?

- PWM = Pulse Width Modulation
- Digital pins can only be fully ON or fully OFF...
- ...but switching ON/OFF very fast can simulate "in-between" brightness
- The ratio of ON-time to OFF-time = duty cycle



PWM vs. True Analog Output

- True analog = a continuously variable voltage (Uno cannot output this)
- PWM = still just HIGH/LOW, but rapidly switched to appear variable
- Uno PWM pins are marked with a ~ symbol (e.g. 3, 5, 6, 9, 10, 11)





Function: analogWrite()

sketch.ino

```
analogWrite(9, 128); // value from 0 (off) to 255 (full brightness)
```

- Must be used on a ~ PWM pin
- 0 = always off, 255 = always on, 128 $\approx$ 50% brightness



Fade LED - Circuit & Code

Same circuit as the single Blink LED, but connected to a ~ pin (e.g. pin 9).

sketch.ino

```
int brightness = 0;

void setup() {
  pinMode(9, OUTPUT);
}

void loop() {
  analogWrite(9, brightness);
  brightness = brightness + 5;
  if (brightness > 255) {
    brightness = 0;
  }
  delay(30);
}
```



Build the Fade LED

5

TASK

Wire the LED to a PWM pin and upload the fade code.

STRETCH GOALS

- Make it fade up and down smoothly, with no snap reset (needs a direction flag)



Pick One, Build It With Your Group

1

Night Lamp

LED fades on gradually, like a bedside lamp

2

Blink Patterns

Creative custom LED sequences — heartbeat, alarm

3

Reaction Timer

LED lights after a random delay; a button press measures reaction speed

4

Morse Code LED

Blink your name or a short word in Morse code



Night Lamp

- Reuse the Fade LED circuit and code
- Add a button to turn the "lamp" on/off entirely
- Slower fade-in speed feels more "lamp-like" when button is pushed



Blink Patterns

- Use multiple digitalWrite + delay calls in sequence to design a pattern
- Ideas: heartbeat (short-short-pause), emergency flash, countdown blink



Reaction Timer

sketch.ino

```
// Rough logic outline:  
// 1. Wait a random delay (delay + random())  
// 2. Turn LED on  
// 3. Start measuring time  
// 4. Wait for button press  
// 5. Measure elapsed time -> print to Serial Monitor
```

- Introduces random() and, optionally, millis() for timing
- Needs Serial.begin() / Serial.println() to show the result



Morse Code LED

- Dot = short blink (~200 ms), Dash = long blink (~600 ms)
- Gap between symbols, longer gap between letters
- Encode a short word — your name, or "HI"

A	● —	U	● ● —
B	— ● ● ●	V	● ● ● —
C	— ● — ●	W	● — —
D	— ● ●	X	— ● ● —
E	●	Y	— ● — —
F	● ● — ●	Z	— — ● ●
G	— — ●		
H	● ● ● ●		
I	● ●		
J	● — — —		
K	— ● —		
L	● — ● ●		
M	— —		
N	— ●		
O	— — —		
P	● — — ●		
Q	— — ● —		
R	● — ●		
S	● ● ●		
T	—		
		1	● — — —
		2	● ● — —
		3	● ● ● —
		4	● ● ● ● —
		5	● ● ● ● ●
		6	— ● ● ● ●
		7	— — ● ● ●
		8	— — — ● ●
		9	— — — — ●
		0	— — — — —



What We Covered Today

Digital output — pinMode, digitalWrite, delay

Digital input — digitalRead, INPUT_PULLUP

PWM — analogWrite, duty cycle

Built: Blink, Traffic Light, Button LED, Fade LED, + a mini project

