

# Ardunio Setup

## Lecture -1

# Arduino Uno Features

1. **More frequency and number of instructions per cycle:** Atmega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions/cycle.
2. **Built-in regulation:** This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
3. **Flexibility & Ease of use:** There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins.
4. **Configurable pins:** The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high range using `analogReference()` function and AREF pin.

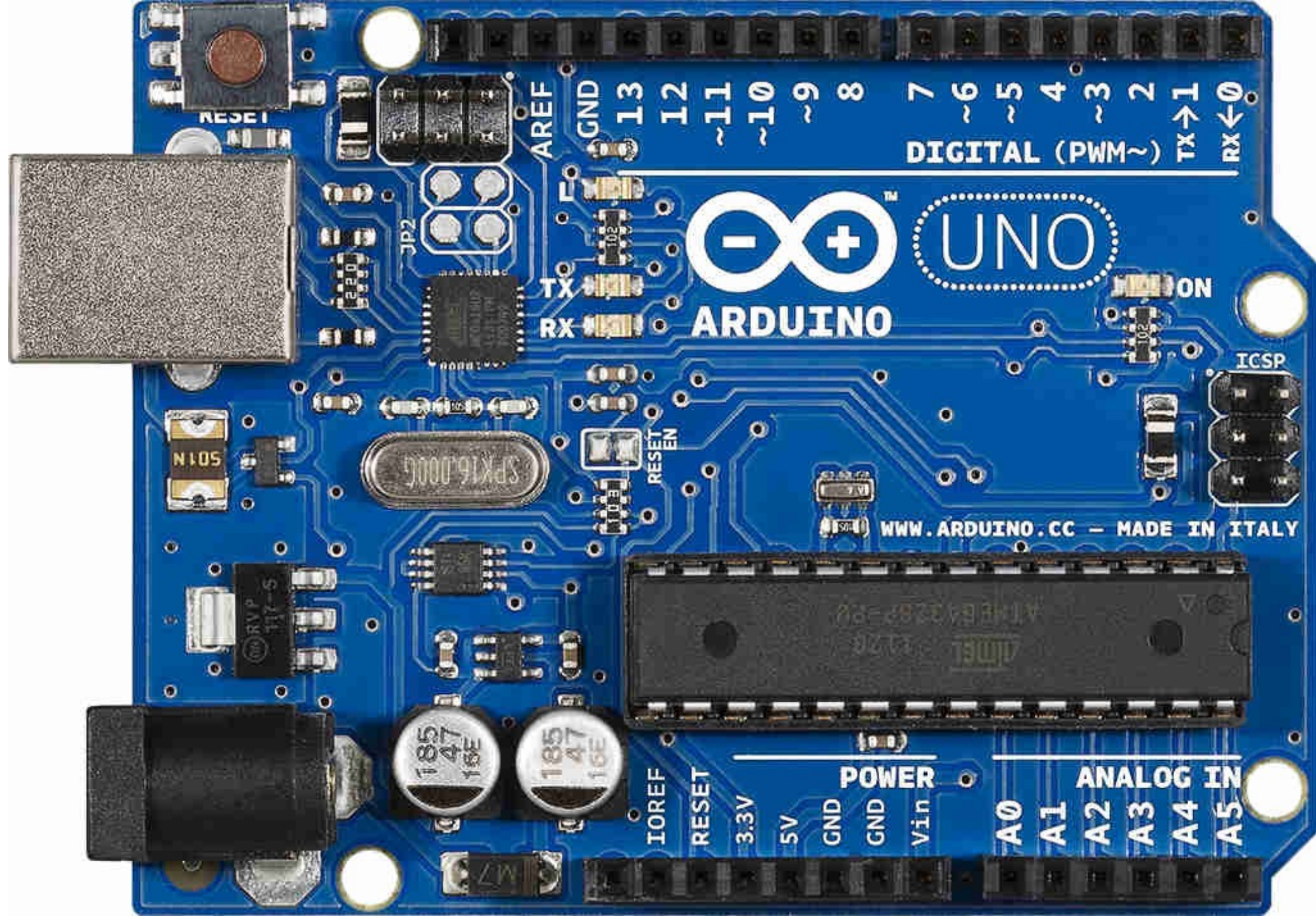
# Arduino Uno Features

5. **Quick Start:** Reset pin is available in the board that reset the whole board and takes the running program in the initial stage. This pin is useful when board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning.
6. **Greater Flash Memory:** 13KB of flash memory is used to store the number of instructions in the form of code.
7. **Low Voltage Requirement:** Only 5 V is required to turn the board on, which can be achieved directly using USB port or external adapter, however, it can support external power source up to 12 V which can be regulated and limit to 5 V or 3.3 V based on the requirement of the project.

# Arduino Uno Features

- 8. **Plug & Play:** There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header.
- 9. **USB interface:** Arduino Uno comes with USB interface i.e. USB port is added on the board to develop serial communication with the computer.
- 10. **Power alternatives:** Apart from USB, battery or AC to DC adopter can also be used to power the board.
- 11. **More Storage:** There is a provision of Mirco SD card to be used in the boards to make them store more information.

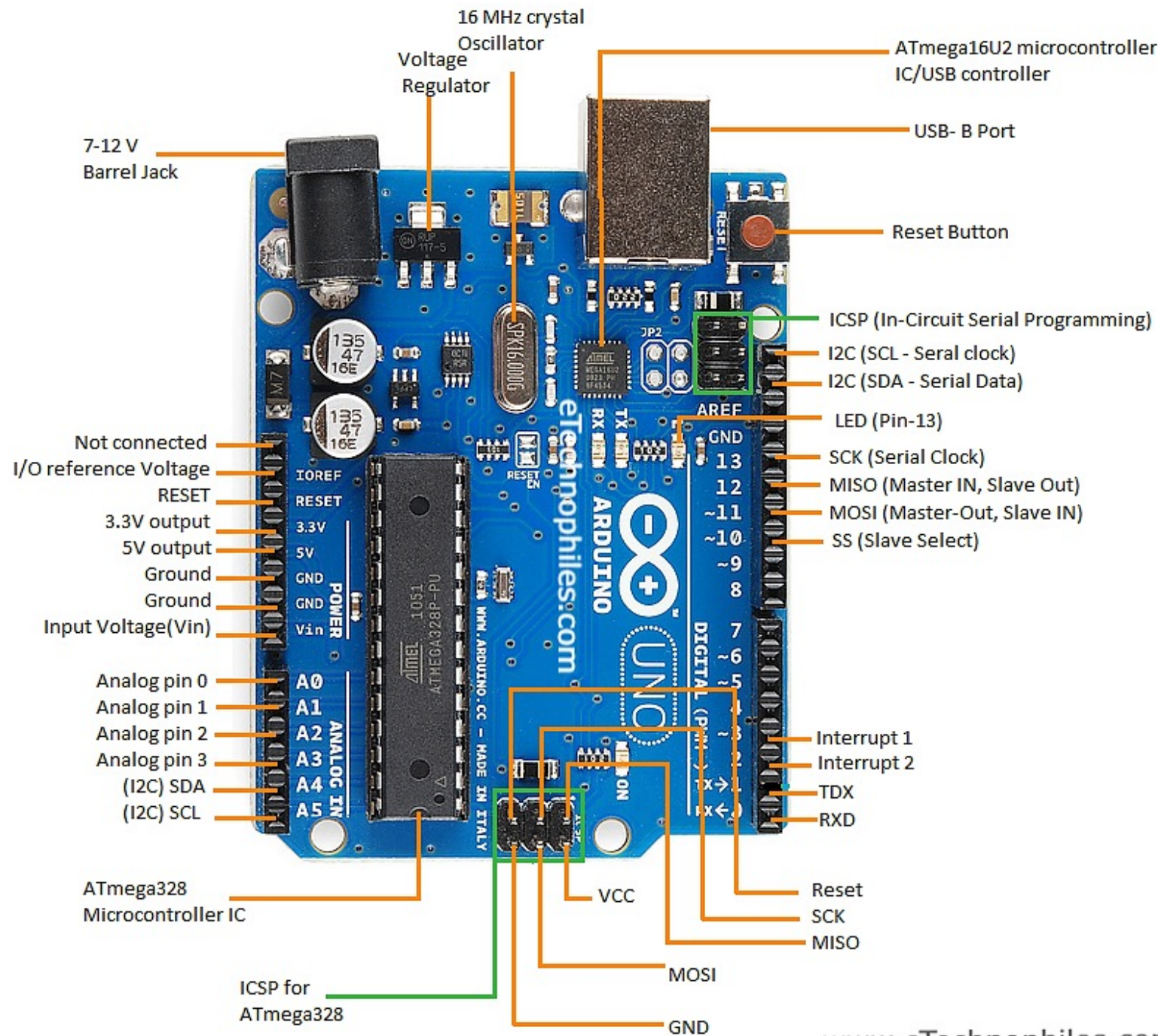




# USB 2.0 Cable Type A/B







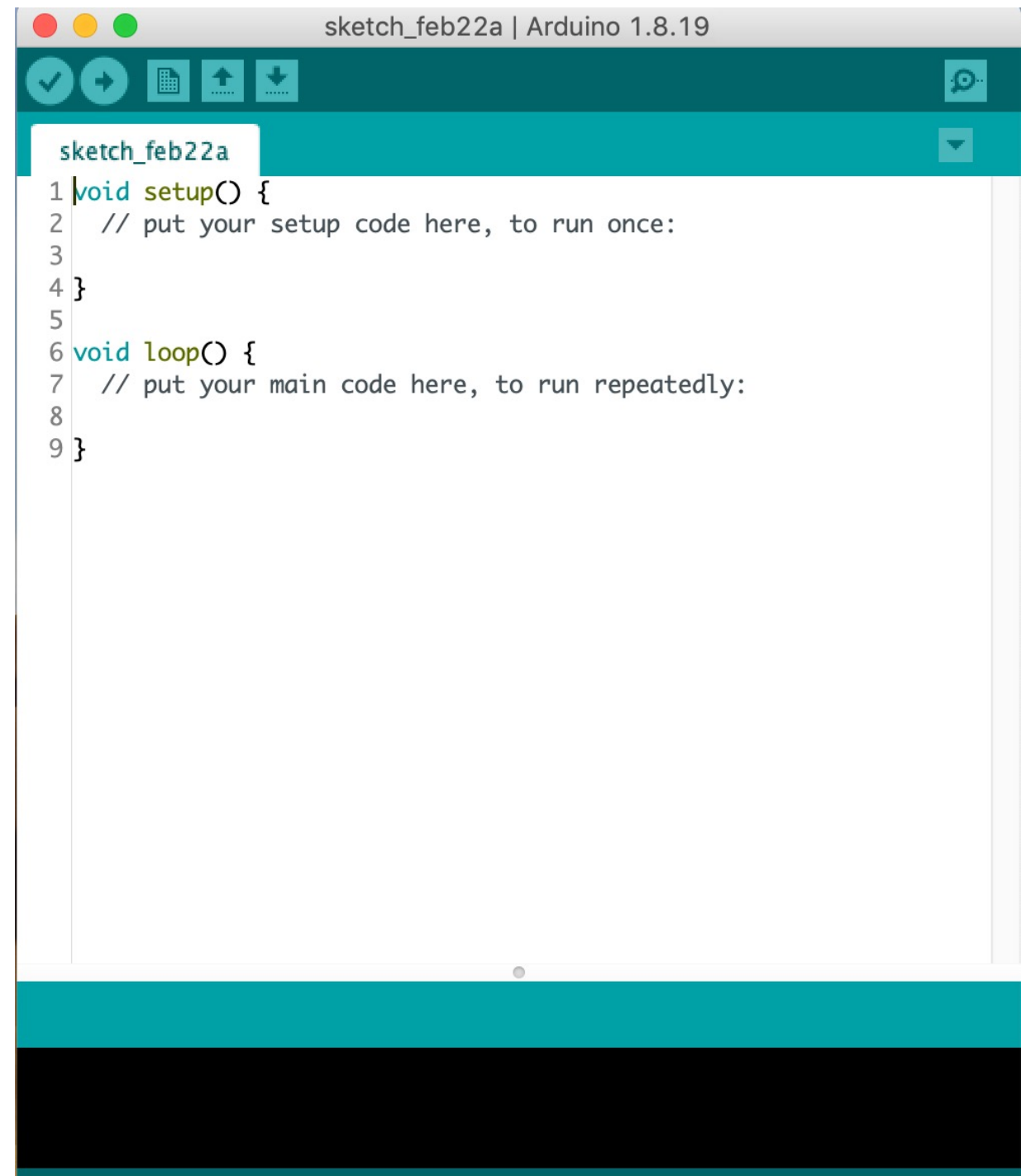
# Arduino Uno Specifications

The specifications of Arduino Uno is as given in the table below.

Microcontroller	ATmega38P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0-A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40mA
DC Current on 3.3V Pin	50mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2kB
EEPROM	1kB
Frequency (Clock Speed)	16MHz



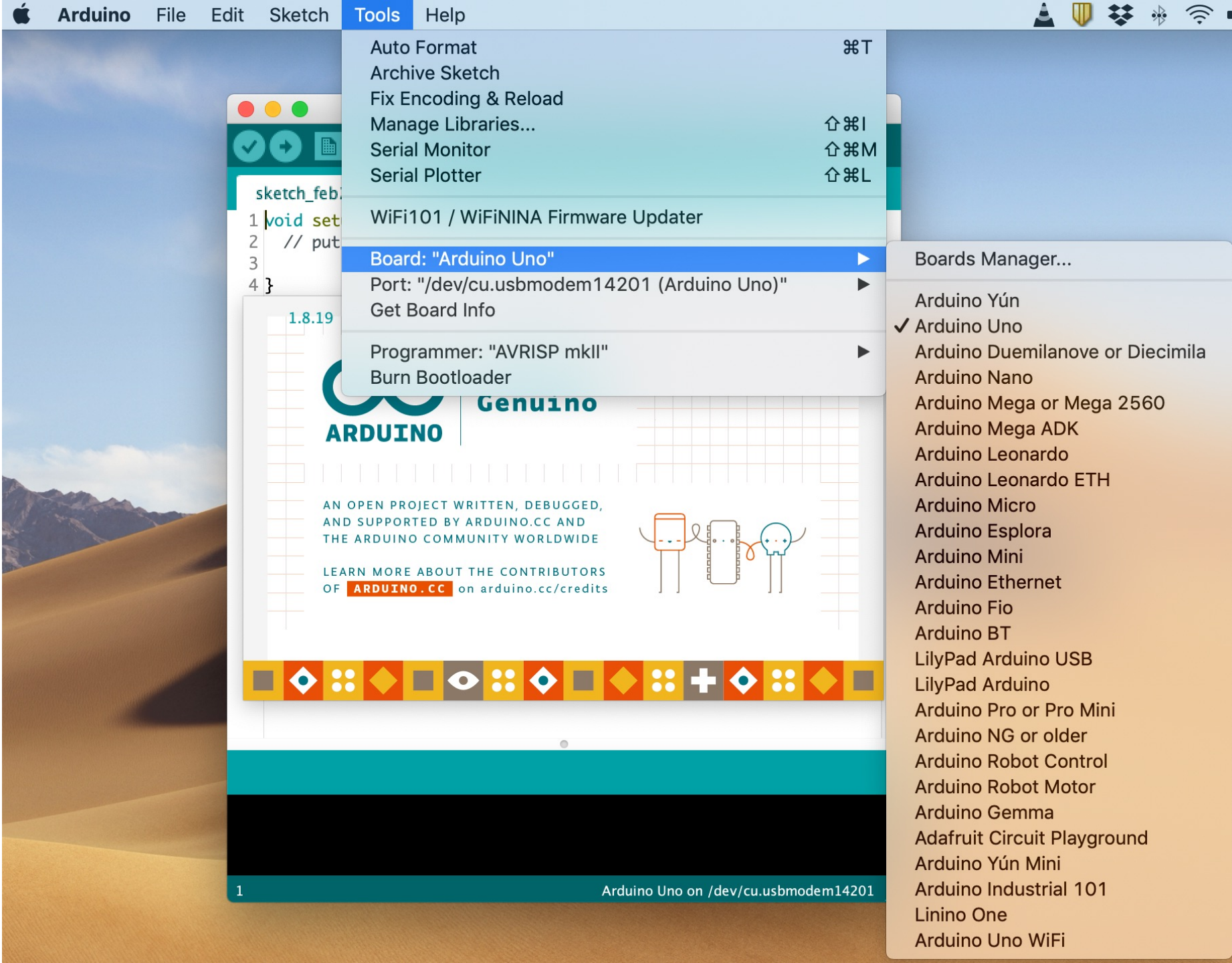
# Ardunio IDE

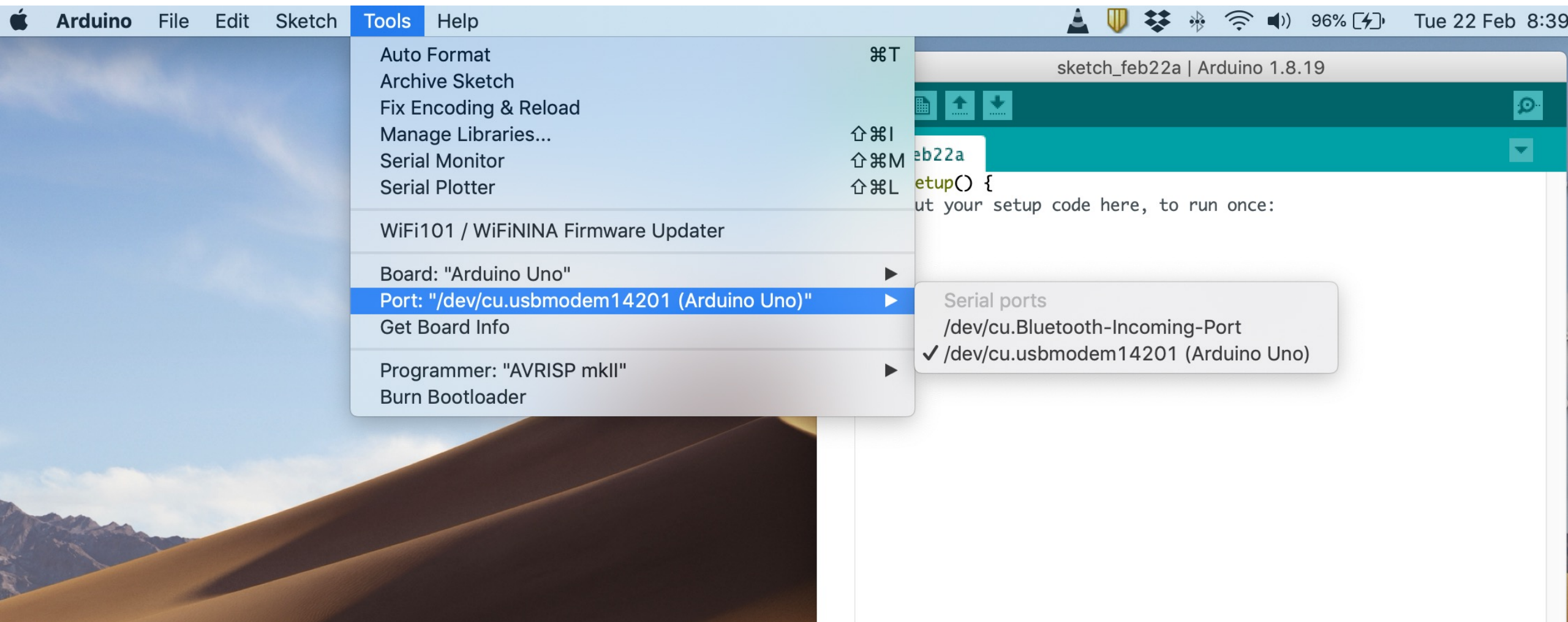


The screenshot displays the Arduino IDE environment. The title bar at the top indicates the file name 'sketch\_feb22a' and the software version 'Arduino 1.8.19'. Below the title bar is a toolbar with icons for checking code, running, saving, and uploading. The main text area contains the following C++ code:

```
sketch_feb22a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```


The bottom of the IDE window features a dark status bar and a black console area for output.







# Blink a Light

The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads "sketch\_feb22a | Arduino 1.8.19". Below the title bar is a toolbar with icons for checking, running, saving, and uploading. The main text area contains a C++ sketch for blinking an LED. The sketch is named "sketch\_feb22a" and is displayed in a monospaced font with syntax highlighting. The code defines a pin number, a delay time, and two functions: setup() and loop(). The setup() function configures the pin as an output. The loop() function writes HIGH to the pin, delays for 10,000 milliseconds, writes LOW to the pin, and delays for another 10,000 milliseconds.

```
sketch_feb22a
1 int ledPin = 13;
2   int delayTime = 10000;
3
4 void setup() {
5   // put your setup code here, to run once:
6   pinMode(ledPin, OUTPUT);
7
8 }
9
10 void loop() {
11   // put your main code here, to run repeatedly:
12   digitalWrite(ledPin, HIGH);
13   delay(delayTime);
14   digitalWrite(ledPin, LOW);
15   delay(delayTime);
16
17 }
18
```

# Morse Code

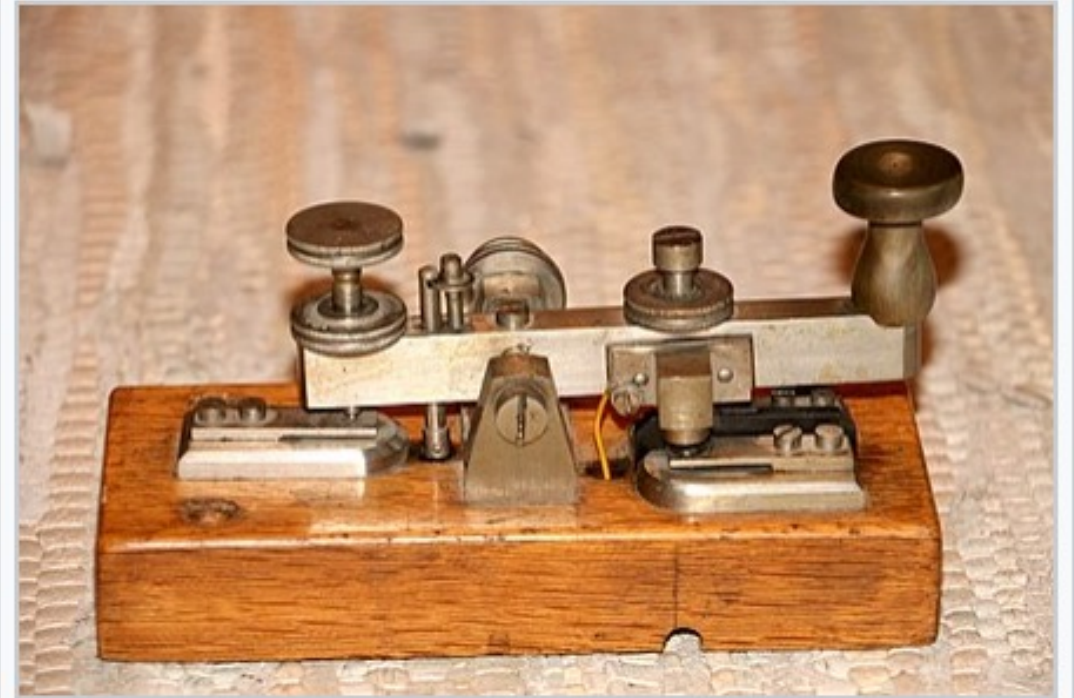
- **Morse code** is a method used in [telecommunication](#) to [encode text](#) characters as standardized sequences of two different signal durations, called ***dots and dashes, or dits and dahs***.<sup>[3][4]</sup> Morse code is named after [Samuel Morse](#), one of the inventors of the [telegraph](#).


# Morse Code

A	• —
B	— • • •
C	— • — •
D	— • •
E	•
F	• • — •
G	— — •
H	• • • •
I	• •
J	• — — —
K	— • — —
L	• — • •
M	— —
N	— •
O	— — —
P	• — — •
Q	— — • —
R	• — •
S	• • •
T	—

U	• • —
V	• • • —
W	• — —
X	— • • —
Y	— • — —
Z	— — • •

1	• — — — —
2	• • — — —
3	• • • — —
4	• • • • —
5	• • • • •
6	— • • • •
7	— — • • •
8	— — — • •
9	— — — — •
0	— — — — —



This Morse key was originally used  by [Gotthard railway](#), later by a [shortwave radio amateur](#)<sup>[2]</sup>



# SOS – Save our Ships

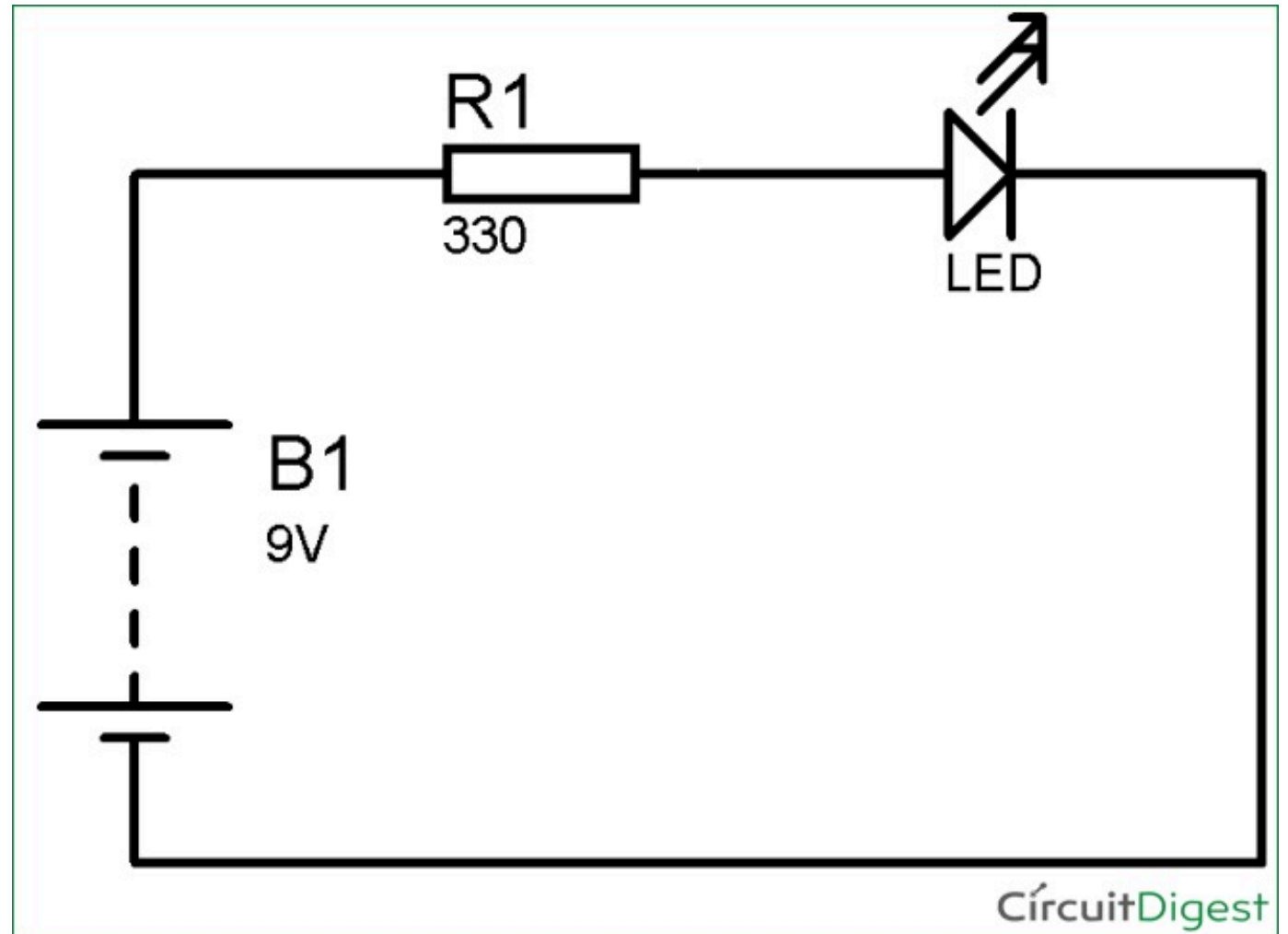
**SOS** is a [Morse code distress signal](#)

(  ), used internationally, that was originally established for maritime use.

```
1 int ledPin=13;
2
3 void setup() {
4   // put your setup code here, to run once:
5   pinMode(ledPin,OUTPUT);
6 }
7
8 void loop() {
9   // put your main code here, to run repeatedly:
10
11   flash(200); flash(200); flash(200); // S
12   delay(300); // otherwise the flashes run together
13   flash(500); flash(500); flash(500); // 0
14   flash(200); flash(200); flash(200); // S
15   delay(1000);
16
17 }
18
19 void flash(int duration)
20 {
21   digitalWrite(ledPin,HIGH);
22   delay(duration);
23   digitalWrite(ledPin,LOW);
24   delay(duration);
25 }
```

# Blink an External LED

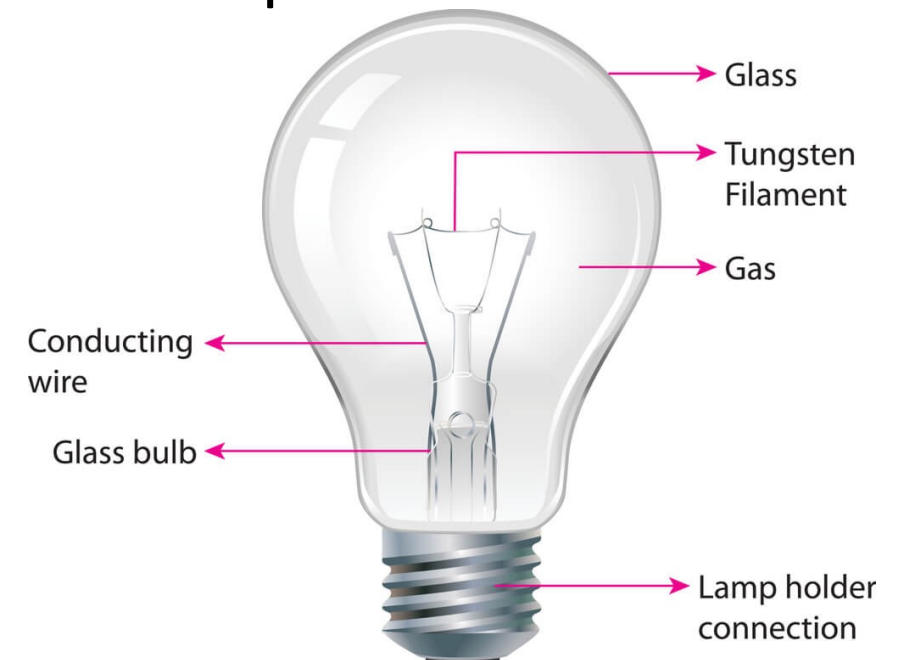
- LED
- Resistors
- Bread Board
- Jumper Wires



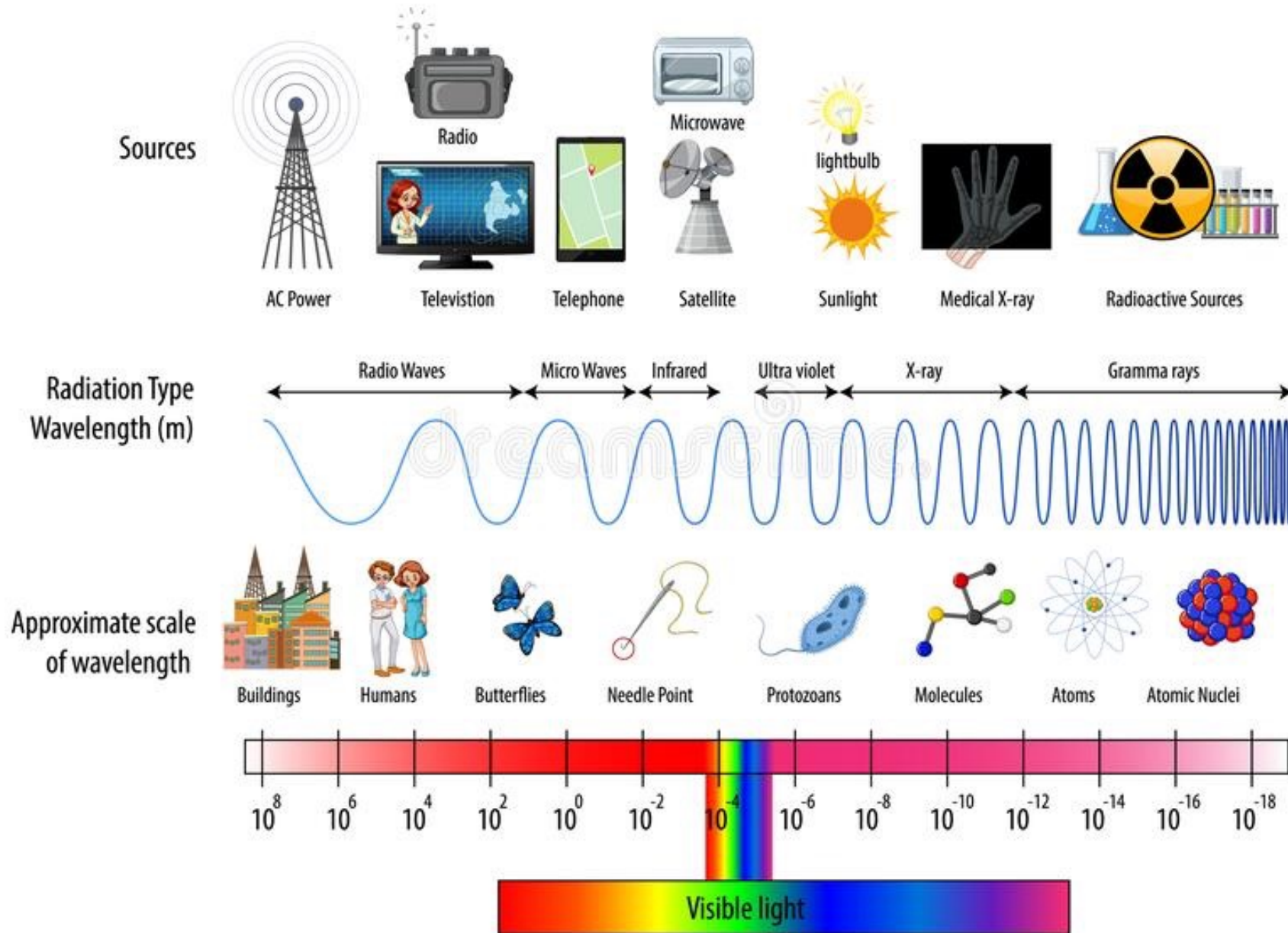


# How a Light bulb works?

- Light is a form of energy that can be released by an atom.
- Atoms release light photons when their **electrons** become excited.
- The **wavelength** of the emitted light (which determines its color) depends on how much energy is released, which depends on the particular position of the electron.
- Heat bulb (95% of energy as heat)



# THE ELECTROMAGNETIC SPECTRUM

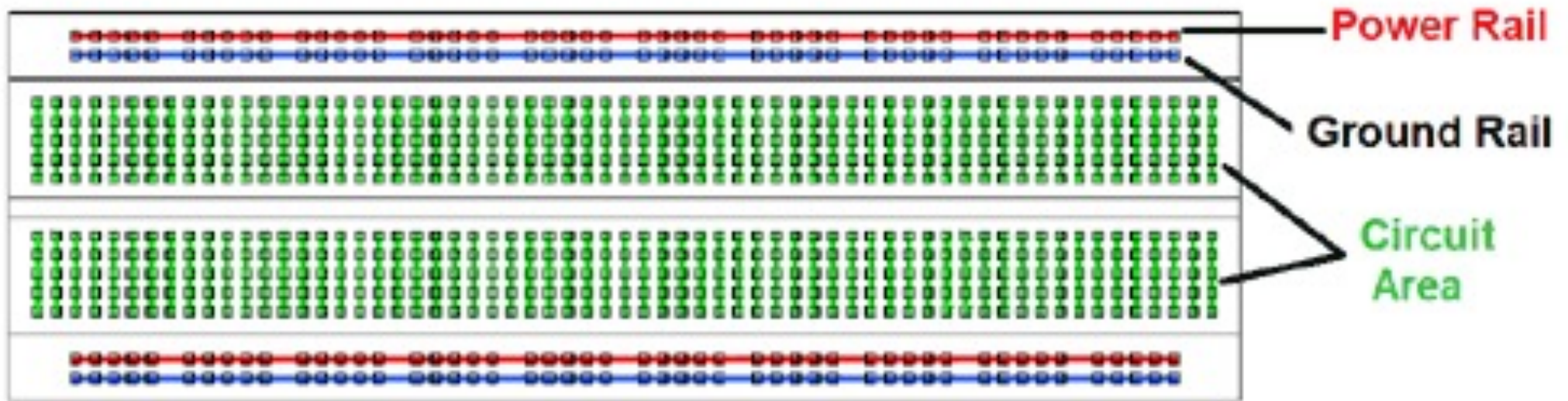


# What is LED

- Light emitting diode – cooler
- Semi-Conductor material
- Most semiconductors are made of a poor conductor that has had impurities
- The process of adding impurities is called **doping**.
- A semiconductor with extra electrons is called **N-type material**
- A semiconductor with extra holes is called **P-type material**



# What is a Bread Board?



# Jumper Wires



# Resistors

