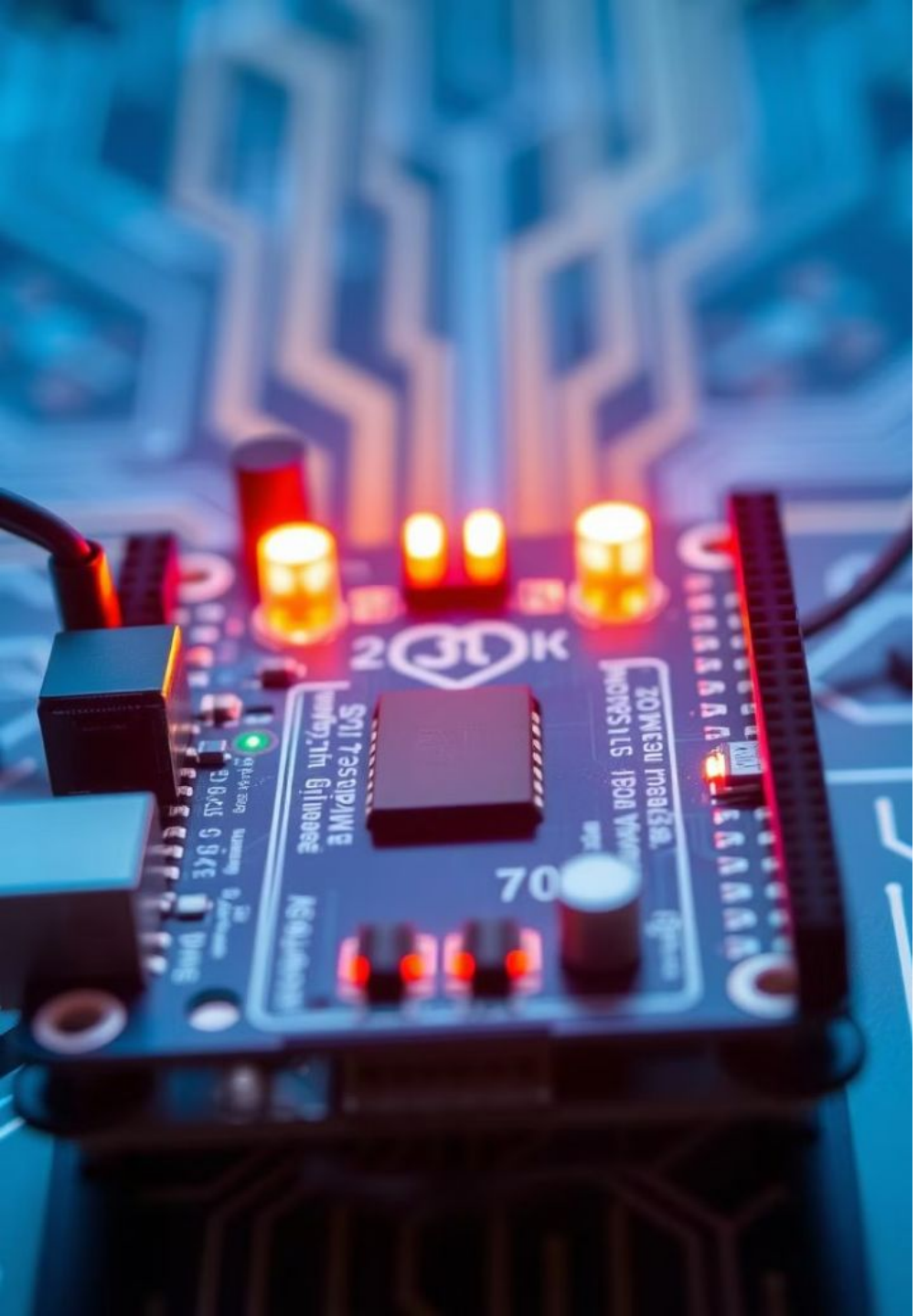


Unlock the Power of Arduino

Welcome to the world of creating with Arduino!

by Dr. Manoj Kumar Sharma

What is Arduino?

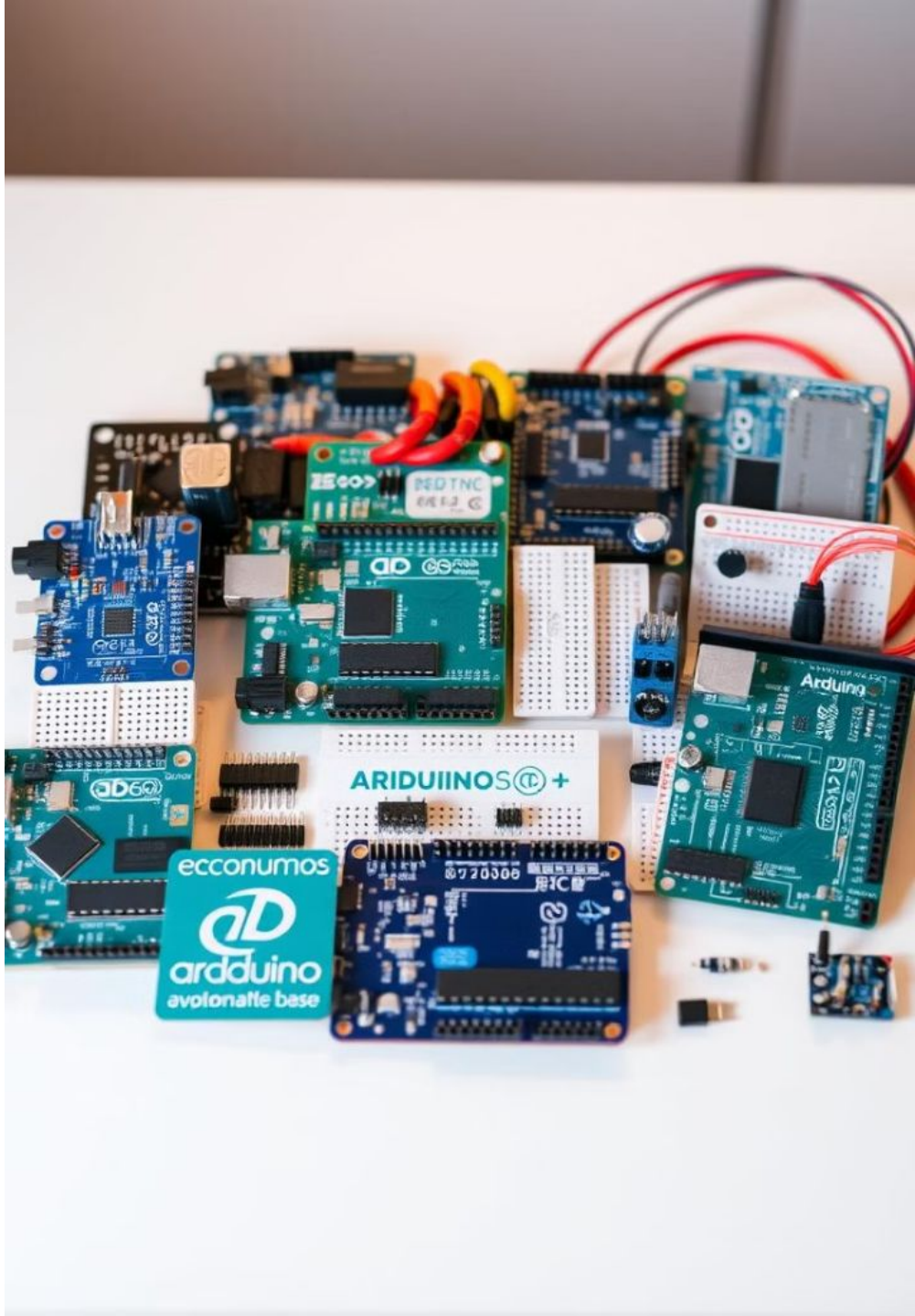
Arduino is an open-source electronics platform.

Hardware

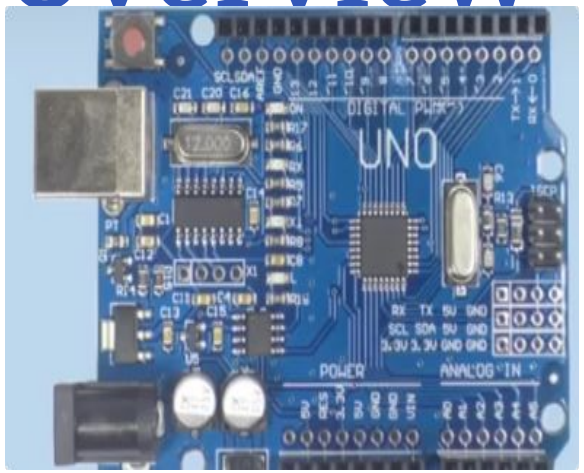
Microcontrollers for building projects.

Software

IDE for programming Arduino boards.

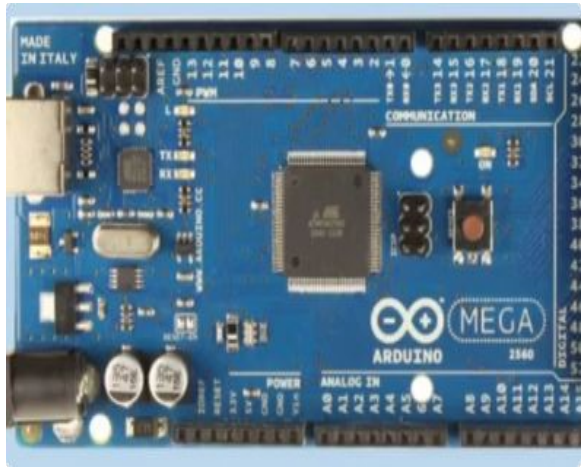


Arduino Hardware Overview



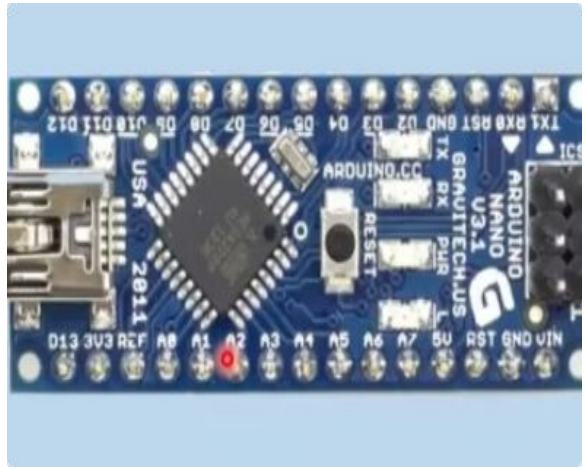
Arduino Uno

A popular, beginner-friendly board. It features a microcontroller, USB port, and various input/output pins for projects.



Arduino Mega

This board has more memory, I/O pins, and processing power, making it suitable for more complex projects.



Arduino Nano

Perfect for space-constrained projects. It's versatile and can be used in a wide range of applications.

WeMos D1

The **WeMos D1** is an ESP8266-based Wi-Fi development board configurable via Arduino IDE for IoT applications.

Arduino Hardware Overview

Explore Arduino's core components.

1. ## GPIO

   General Purpose Input/Output pins.

2. ## Communication

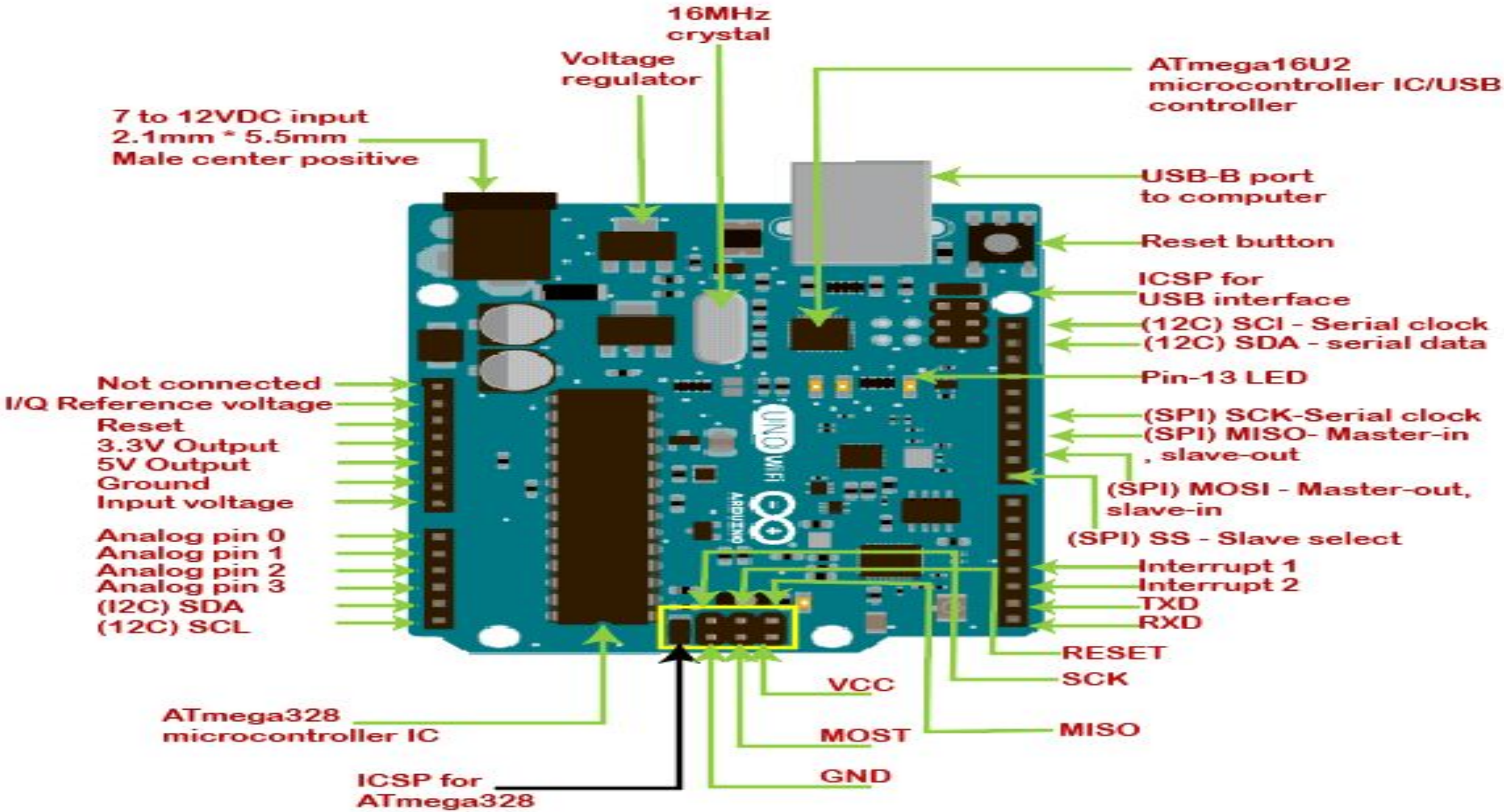
   UART, I2C, and SPI protocols.

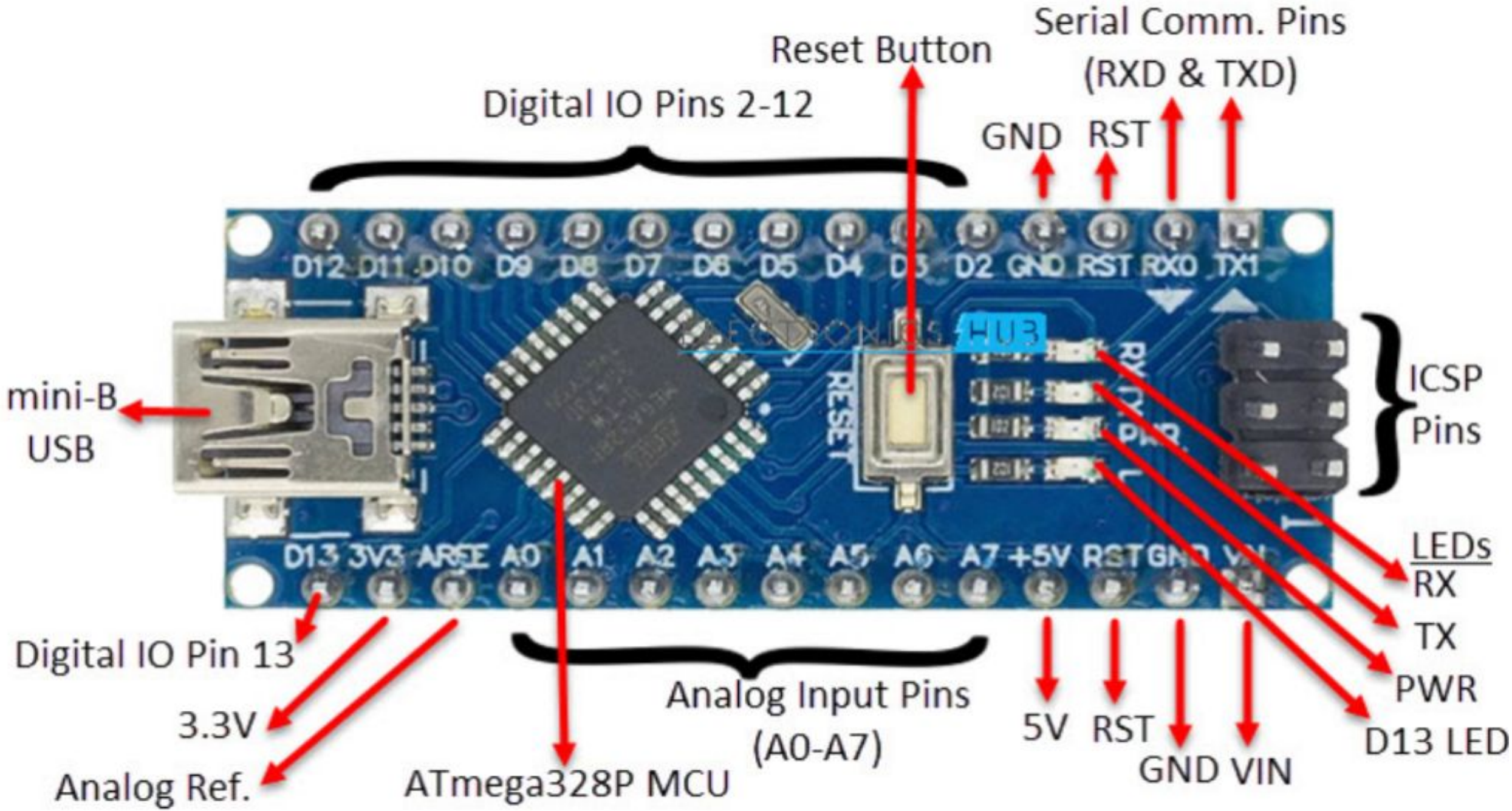
3. ## Clocks and Timers

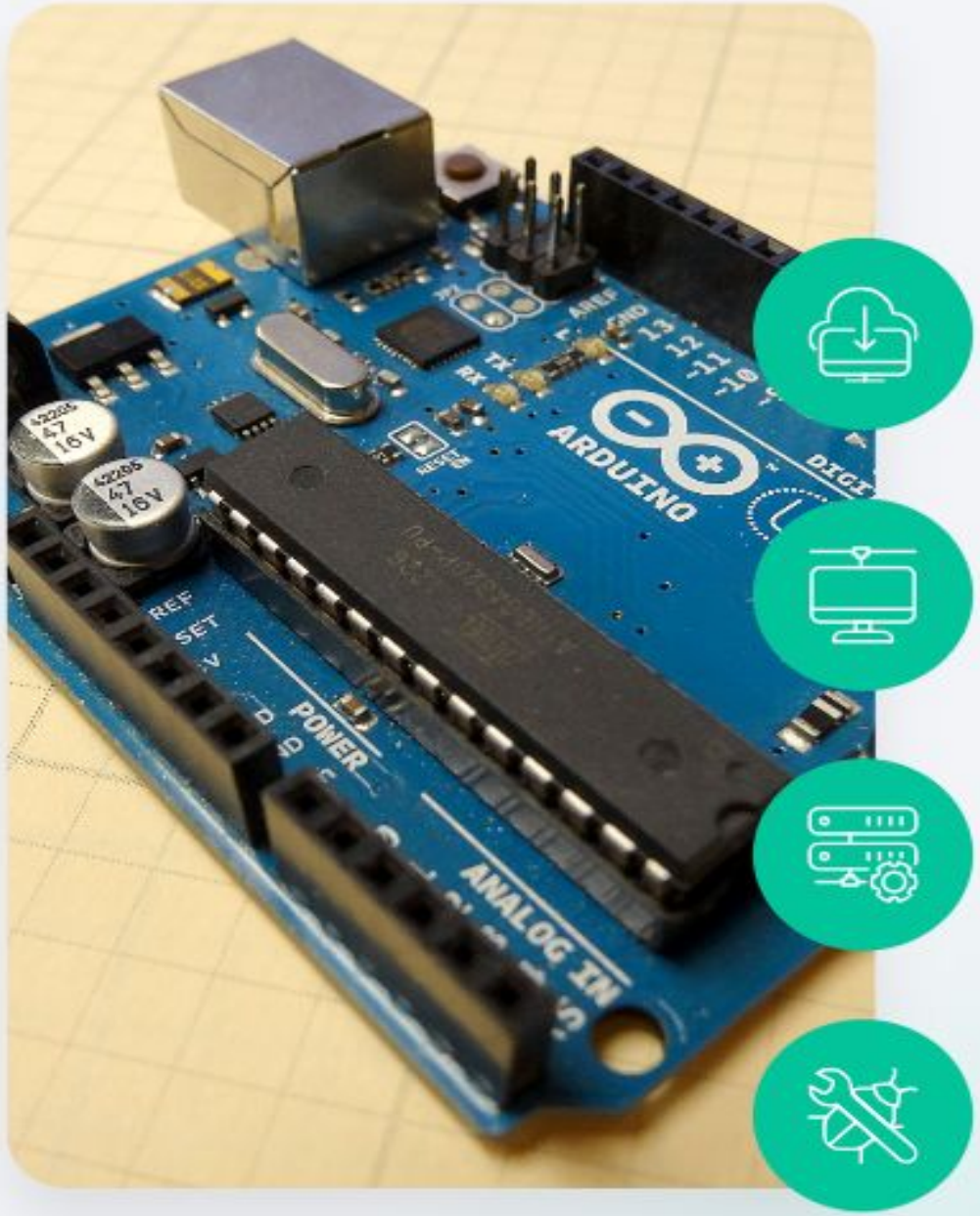
   For timing events and controlling processes.

4. ## PWM

   Pulse Width Modulation for controlling analog devices.


16MHz crystal
Voltage regulator
ATmega16U2 microcontroller IC/USB controller
7 to 12VDC input
2.1mm * 5.5mm
Male center positive
USB-B port to computer
Reset button
ICSP for USB interface
(12C) SCl - Serial clock
(12C) SDA - serial data
Pin-13 LED
Not connected
I/Q Reference voltage
Reset
3.3V Output
5V Output
Ground
Input voltage
(SPI) SCK-Serial clock
(SPI) MISO- Master-in , slave-out
(SPI) MOSI - Master-out, slave-in
(SPI) SS - Slave select
Analog pin 0
Analog pin 1
Analog pin 2
Analog pin 3
(I2C) SDA
(12C) SCL
Interrupt 1
Interrupt 2
TXD
RXD
RESET
SCK
VCC
MOST
MISO
GND
ATmega328 microcontroller IC
ICSP for ATmega328


Reset Button
Serial Comm. Pins
(RXD & TXD)
Digital IO Pins 2-12
GND
RST
mini-B
USB
ICSP
Pins
LEDs
RX
TX
PWR
D13 LED
Digital IO Pin 13
3.3V
Analog Ref.
ATmega328P MCU
Analog Input Pins
(A0-A7)
5V
RST
GND
VIN



Setting Up Your Arduino

A Step-by-Step Guide

Software Installation

Download and install the Arduino IDE.

Connecting Hardware

Use a USB cable to connect the Arduino board to your computer.

IDE Configuration

Select the correct board and port in the Arduino IDE.

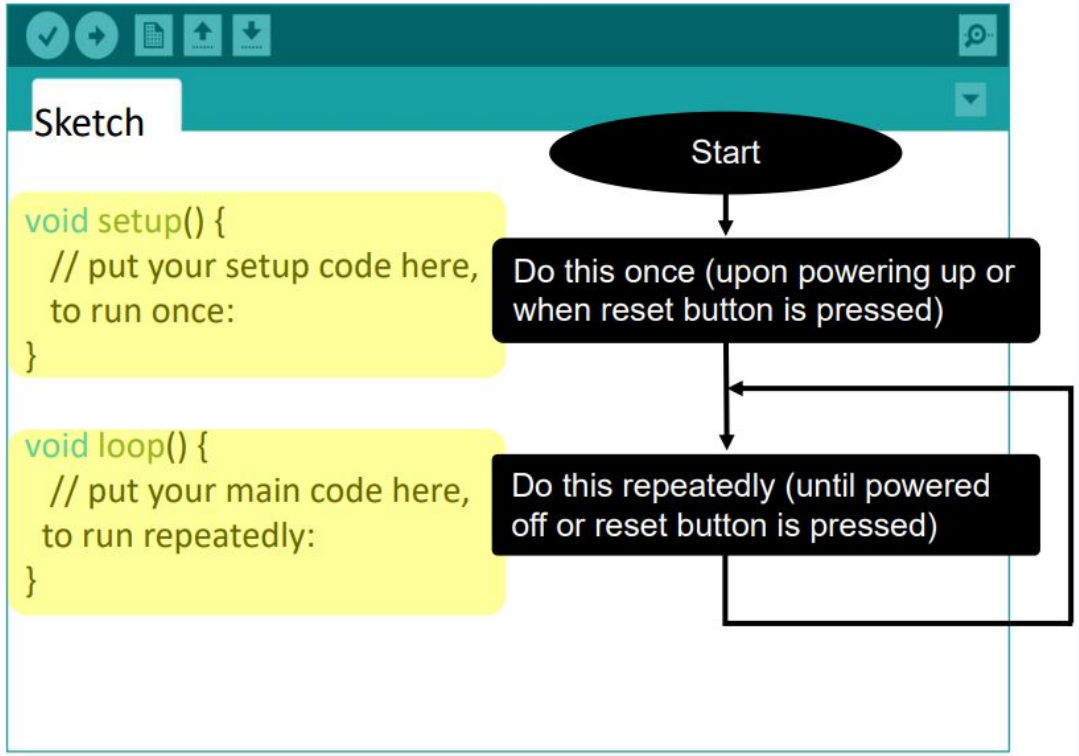
First Program

Write and upload a simple introductory program.

Arduino Programming Basics

Understanding the core concepts of Arduino programming.

void setup()	Initializes your program.
void loop()	Runs continuously after setup.




```
// Blink an LED
void setup() {
  pinMode(LED_BUILTIN, OUTPUT); // Initialize the built-in
}

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

Hands-On: Your First Arduino Program

Blink an LED, a classic Arduino project.

Hardware

Connect LED to a digital pin.

Use a resistor for current limiting.

Code

Use _digitalWrite()_ to toggle the LED.

Use _delay()_ to control blinking speed.

Arduino Sketch

Every sketch has these functions:

- void **setup**()
  - Runs once at the very beginning
  - Set up your variables, peripherals

- void **loop**()
  - Runs forever
  - Code that does actual work goes here

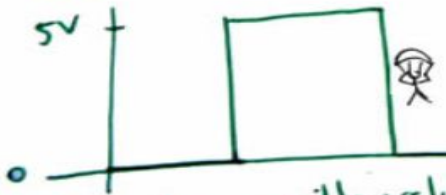
Analog vs Digital

Analog vs Digital

Analog:

Any value between 0 and 5V is possible

Digital:



Only two possible values: 0 & 5V
Also known as 0,1 or LOW, HIGH

Why digital?

- Digital is simple, but lets us use complex math.
- Digital is much less sensitive to noise
- Digital opens up the possibility of software

Why analog?

- Our world is analog!
- Digital is just a special case of analog

Essential Functions

- pinMode()
- digitalWrite()
- digitalRead()
- analogWrite()
- analogRead()
- delay()
- Serial.begin()
- Serial.print()

pinMode() Function

The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

Syntax

```
void setup()
{
pinMode(pin, mode);
}
```

- **pin** – the number of the pin whose mode you wish to set
- **mode** – INPUT, OUTPUT, or INPUT_PULLUP.

digitalWrite() Function

The **digitalWrite()** function is used to write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode(), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.

Syntax

```
void loop()
{
 digitalWrite(pin, value);
}
```

- **pin** – the number of the pin where you wish to set the **value**
- **value** –HIGH or LOW

digitalRead() Function

The **digitalRead()** function is used to write a HIGH or a LOW value to a digital pin. If the pin has been configured as an INPUT with pinMode(), its voltage will be read to the corresponding value: 1 for 5V, 0 for 0V.

Syntax

```
void loop()
{
 digitalRead(pin);
}
```

- **pin** – the number of the pin where you wish to read the value

analogRead( ) function

In the lower-right part of the Arduino board, you will see six pins marked "Analog In". These special pins not only tell whether there is a voltage applied to them, but also its value. By using the **analogRead()** function, we can read the voltage applied to one of the pins.

This function returns a number between 0 and 1023, which represents voltages between 0 and 5 volts. For example, if there is a voltage of 2.5 V applied to pin number 0, analogRead(0) returns 512.

Syntax

```
void loop()
{
 analogRead(pin);
}
```

- **pin** – the number of the pin whose value you wish to read

analogWrite() Function

The **analogWrite()** function writes an analog value (PWM wave) to a pin. After a call of the analogWrite() function, the pin will generate a steady square wave of the specified duty cycle until the next call to analogWrite() or a call to digitalRead() or digitalWrite() on the same pin. The frequency of the PWM signal on most pins is approximately 490 Hz. On the Uno and similar boards, pins 5 and 6 have a frequency of approximately 980 Hz.

Syntax

```
void loop()
{
 analogWrite(pin, value);
}
```

- **pin** – the number of the pin where you wish to set the **value**
- **value** –HIGH or LOW

Serial.begin()

Sets the data rate in bits per second (baud) for serial data transmission. For communicating with the computer, use one of these rates: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, or 115200

Syntax:

```
void setup()
{
Serial.begin(9600);
}
```

Serial.print()

Used to print data onto the serial monitor.
Serial.print will keep cursor on same line after printing
Serial.println will move cursor to the next line after printing
It can work only if Serial.begin() function is present

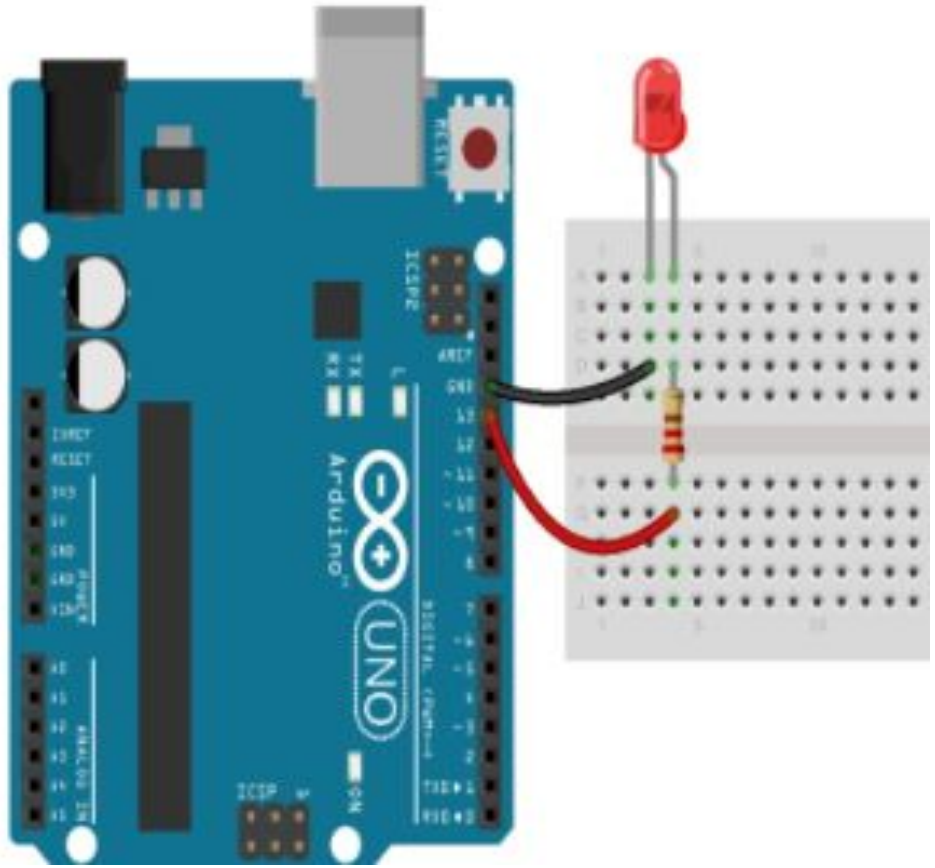
Syntax:

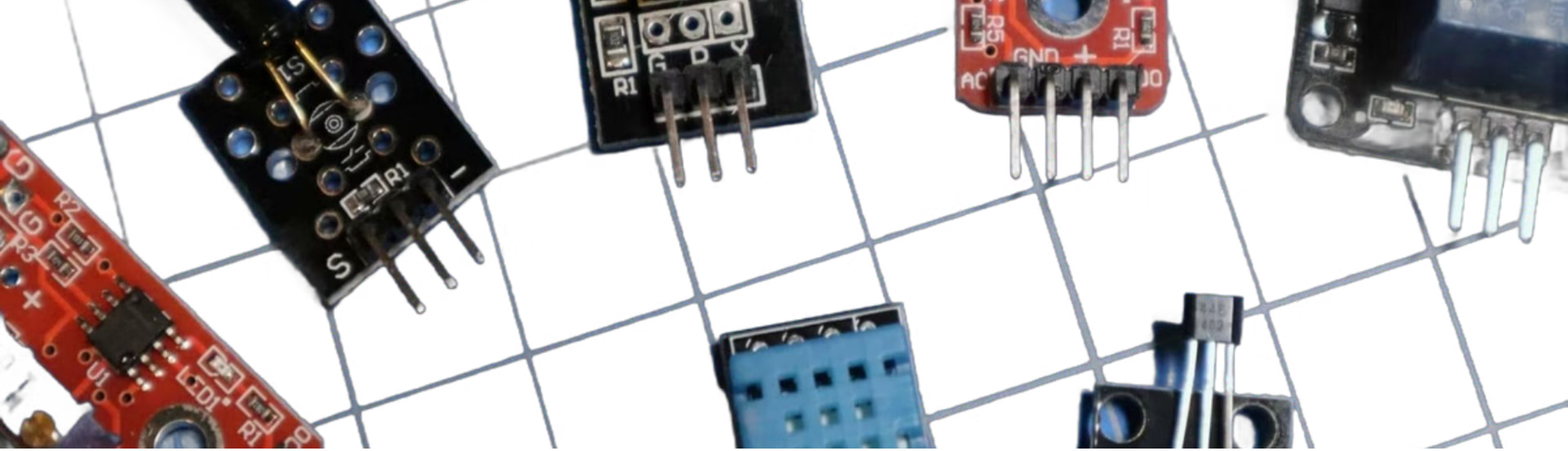
```
void loop()
{
Serial.println (“Hello World”);
}
```

```
// Blink an LED
void setup() {
  pinMode(LED_BUILTIN, OUTPUT); // Initialize the built-in LED pin as an output
}

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

External Led Connection





Hardware Peripherals

Expanding Arduino's capabilities with peripherals.

Overview

LEDs

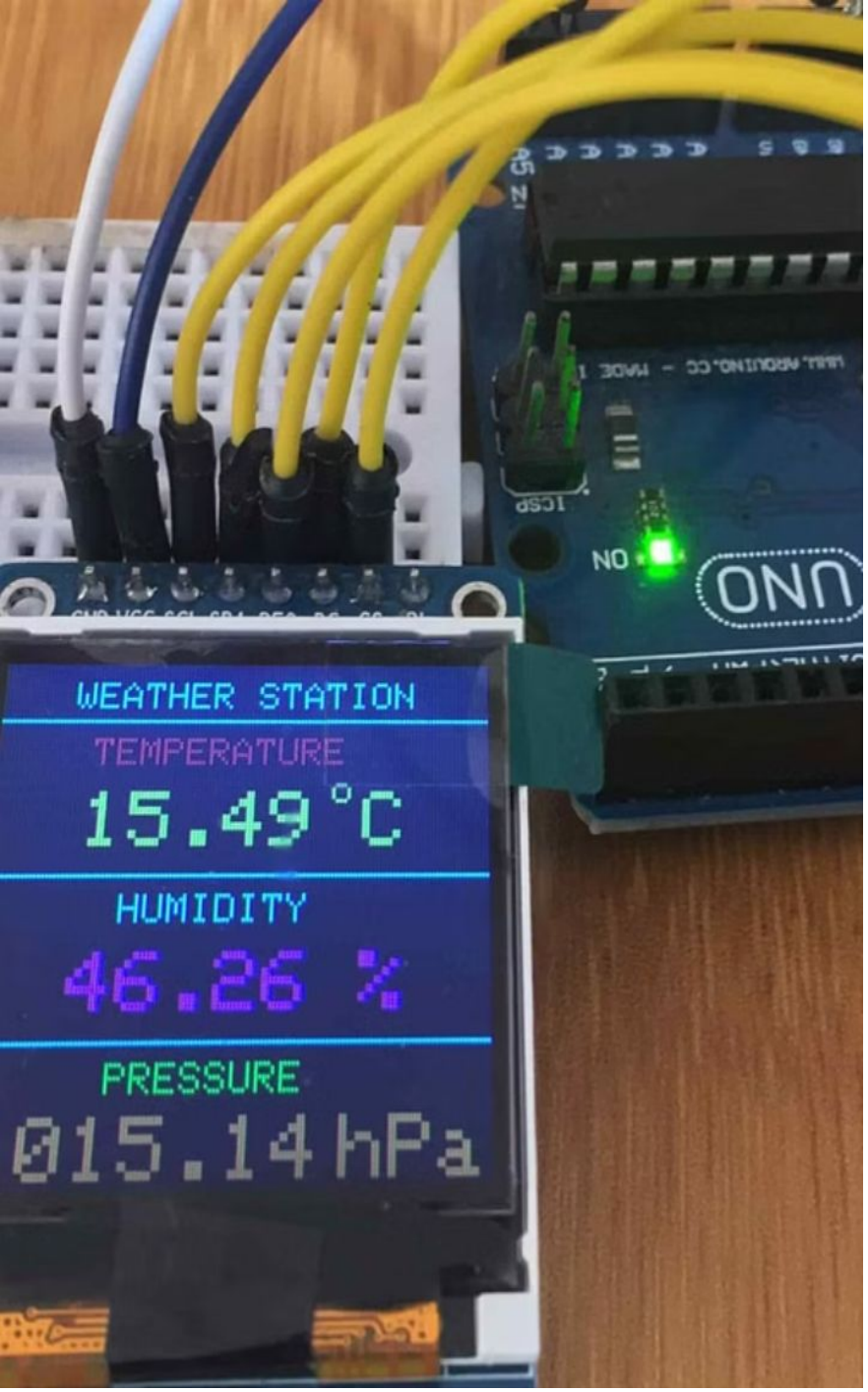
Light-emitting diodes.

Sensors

Measure physical quantities.

Actuators

Control physical devices.



Project 2: Reading

Measure temperature or light intensity.

Sensor Data

1

Connect

Connect the sensor to Arduino.

2

Read

Use _analogRead()_ to get data.

3

Display

Show data on an LCD or serial monitor.

Advanced Concepts: Communication Protocols

Communicate with external devices.

1 **UART**

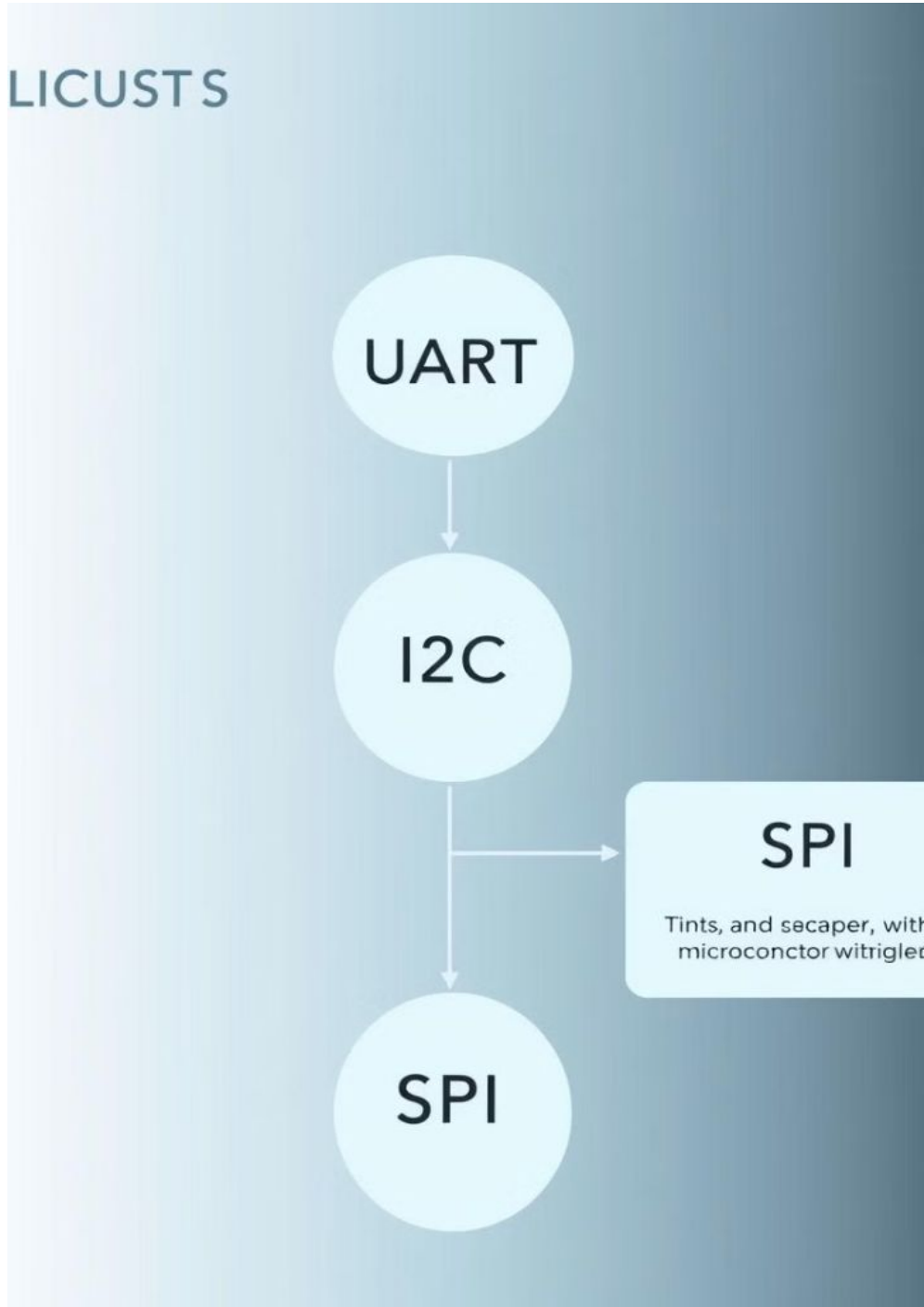
Serial communication for simple data exchange.

2 **I2C**

Two-wire protocol for communication with multiple devices.

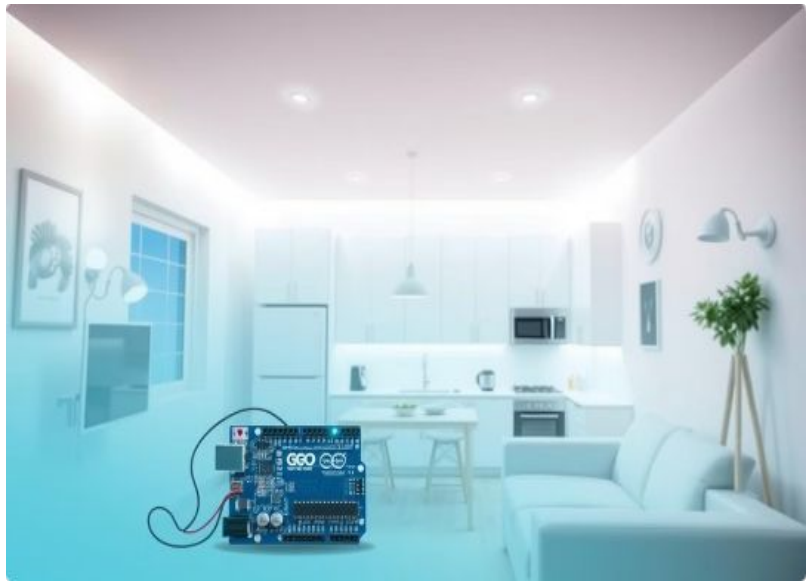
3 **SPI**

High-speed protocol for communication with multiple devices.



Real-World Applications of Arduino

Arduino's potential for practical projects.



Home Automation

Control lights, appliances, and security.

Robotics

Build and control robots.

Environmental Monitoring

Track air quality, water quality, and temperature.

Q&A

This is a dedicated space to address any questions you might have about the Arduino platform and the exciting possibilities.