Arduino C++

Coding / Uno / Nano

The core I/O calls and a non-blocking timing pattern you can reuse.

01 Core API

Setup
setup() runs once; loop() repeats. pinMode(pin, OUTPUT / INPUT / INPUT_PULLUP).

Digital
digitalWrite(pin, HIGH/LOW); digitalRead(pin). Pull-up buttons read LOW when pressed.

Analog
analogRead(A0): 0-1023 on classic Uno/Nano. analogWrite(PWMpin, 0-255).

Serial
Serial.begin(9600); Serial.println(value); match the monitor baud rate.

02 Timing

millis()
Milliseconds since boot as unsigned long. Compare elapsed time by subtraction to handle rollover.

PWM
analogWrite is PWM on Uno/Nano, not a true analog voltage. Use a pin marked ~.

Portability
ADC/PWM resolution and supported APIs vary by board core.

Code example / cpp

Arduino C++ / continued

Coding / Uno / Nano

```
const byte led = LED_BUILTIN;
unsigned long previous = 0;
bool state = false;

void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  unsigned long now = millis();
  if (now - previous >= 500UL) {
    previous = now;
    state = !state;
    digitalWrite(led, state);
  }
}
```

Before you power on

Long delay() calls block sensing and control. Use elapsed-time checks when multiple tasks must run.

Datasheets & further reading

Arduino language reference

https://docs.arduino.cc/language-reference/