Understanding digitalWrite in setup vs. loop

Introduction

In Arduino programming, the setup and loop functions serve distinct roles. The setup function is executed once at the beginning of your program, while the loop function runs continuously. This distinction is crucial when understanding the placement of digitalWrite statements.

digitalWrite in setup

When you place digitalWrite statements within the setup function, you're essentially setting the initial state of your digital pins. In your example, the following lines in setup ensure that the LEDs connected to pins 9, 8, and 7 are turned on immediately after the Arduino board starts running:

C++

```
digitalWrite(ledPin9, HIGH);
digitalWrite(ledPin8, HIGH);
digitalWrite(ledPin7, HIGH);
```

digitalWrite in loop

Placing digitalWrite statements within the loop function allows you to control the state of your digital pins dynamically. In your example, the loop function is repeatedly setting the LEDs to HIGH, effectively keeping them turned on continuously:

C++

```
digitalWrite(ledPin9, HIGH);
digitalWrite(ledPin8, HIGH);
digitalWrite(ledPin7, HIGH);
```

Why Similar Output with Either Placement in This Case

In your specific example, you're not changing the LED states within the loop function. Therefore, the LEDs will remain on regardless of whether you place digitalWrite in setup or loop. However, the distinction becomes clearer when you want to control the LED states dynamically.

Example: Blinking LEDs

To demonstrate the difference, let's modify the code to make the LEDs blink:

C++

```
void loop() {
  digitalWrite(ledPin9, HIGH);
  digitalWrite(ledPin8, HIGH);
  digitalWrite(ledPin7, HIGH);
  delay(1000); // Wait for 1 second
  digitalWrite(ledPin9, LOW);
  digitalWrite(ledPin8, LOW);
  digitalWrite(ledPin7, LOW);
  delay(1000); // Wait for 1 second
}
```

In this case, the LEDs will turn on for 1 second, then turn off for 1 second, creating a blinking effect. This dynamic behavior is achieved by placing the digitalWrite statements within the loop function.

Summary

- **setup:** Used for one-time initialization of pins and setting initial states.
- **loop:** Used for continuous actions and controlling the state of pins dynamically.

By understanding the roles of setup and loop, you can effectively control the behavior of your digital pins in Arduino projects.