Programming Tiny, Colorful Computers

Stephen A. Edwards

March 24, 2018
The Arduino Nano

- Mini USB Jack
- Atmel Mega328P Microcontroller
- Reset Switch
- Receive and Transmit LEDs
- Power Light
- User-controlled LED
Getting Started

Start the Arduino IDE
Select Tools → Board → Arduino Nano
Getting Started Continued

Plug the USB cable into your board

Plug your board into your computer

*The board’s power light should be on*

Select Tools → Port → COM3

*Which COM port may vary; choose the one that is there*
Select File → Examples → 01.Basics → Blink
Upload the Sketch to the Board

Should say “Done uploading.” The user LED should blink
The Blink Example

// setup() runs once: when power is applied or after reset
void setup() {

    // Set the LED_BUILTIN pin to be an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// loop() runs over and over again forever
void loop() {

digitalWrite(LED_BUILTIN, HIGH); // Turn the LED on ("HIGH")
delay(1000); // Wait for 1000 ms = 1s

digitalWrite(LED_BUILTIN, LOW); // Turn the LED off ("LOW")
delay(1000); // Wait a second
}
Each pin has a number (the pink boxes on “Nano pinout”)  

Most pins can be either inputs or outputs

```c
pinMode(4, OUTPUT);  // Control the voltage on pin 4
pinMode(6, INPUT);   // Observe the voltage on pin 6
pinMode(19, INPUT_PULLUP); // Observe 19; “suggest” it be high
pinMode(LED_BUILTIN, OUTPUT); // Control pin 13, LED “L”
```
The input voltage to the board when it is running from external power. Not USB bus power.
Digital Input and Output

Digital: on or off, high or low voltage; nothing in between

digitalWrite(13, LOW); // Turn off the user LED

digitalWrite(13, HIGH); // Turn on the user LED

A digital read from a pin reports either HIGH or LOW

if ( digitalRead(19) == LOW ) {
   // Low voltage on pin 19, a "0"
}

if ( digitalRead(19) == HIGH ) {
   // High voltage on pin 19, a "1"
}
Connect the RGB LED and switch
Reacting to the switch

const int switchpin = 19; // Marked "A5"

void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(switchpin, INPUT_PULLUP); // HIGH unless pressed
}

void loop() {
    if (digitalRead(switchpin) == LOW) { // Switch pressed?
        digitalWrite(LED_BUILTIN, HIGH); // Yes: light LED
    } else {
        digitalWrite(LED_BUILTIN, LOW); // No: turn off LED
    }
}
Controlling the RGB LED

```cpp
const int redpin = 3, greenpin = 5, bluepin = 6;  // D3, D5, D6

void rgb(int r, int g, int b) {
    analogWrite(redpin, r);  // Set the red brightness
    analogWrite(greenpin, g); // Set the green brightness
    analogWrite(bluepin, b);  // Set the blue brightness
}

void setup() {
    pinMode(redpin, OUTPUT);
    pinMode(greenpin, OUTPUT);
    pinMode(bluepin, OUTPUT);
}

void loop() {
    rgb(10,0,0);  delay(500);
    rgb(0,10,0);  delay(500);
    rgb(0,0,10);  delay(500);
}
```
Mood Lighting Control

```c
const int redpin = 3, greenpin = 5, bluepin = 6;    // D3, D5, D6
const int switchpin = 19;
int red = 0, green = 0, blue = 0;

void rgb(int r, int g, int b) {
    analogWrite(redpin, r);        // Set the red brightness
    analogWrite(greenpin, g);      // Set the green brightness
    analogWrite(bluepin, b);       // Set the blue brightness
}

void setup() {
    pinMode(redpin, OUTPUT);
    pinMode(greenpin, OUTPUT);
    pinMode(bluepin, OUTPUT);
    pinMode(switchpin, INPUT_PULLUP);
    red = green = blue = 0;        // Start with the LED off
    rgb(red, green, blue);
}
```
void loop() {
    while (digitalRead(switchpin) == HIGH) {}
    while (digitalRead(switchpin) == LOW) {
        red = (red + 1) % 12; // Add 1 and wrap around at 12
        rgb(red, green, blue);
        delay(200);
    }
    while (digitalRead(switchpin) == HIGH) {}
    while (digitalRead(switchpin) == LOW) {
        green = (green + 1) % 12;
        rgb(red, green, blue);
        delay(200);
    }
    while (digitalRead(switchpin) == HIGH) {}
    while (digitalRead(switchpin) == LOW) {
        blue = (blue + 1) % 12;
        rgb(red, green, blue);
        delay(200);
    }
}
Challenge Problems

- A traffic light
  Red, green, then briefly yellow
  Click to advance

- Reaction-time game
  Countdown with colors
  Press the button quickly
  Blink or color to indicate your speed

- Better mood light color control
  Cycle through “good” colors
  Separately cycle through brightness

- Morse code practice
  Display a color letter code (Red = R, Green = G, etc.)
  Expect player to enter Morse code for the letter
  Display score with blinks or color