

Arduino programming

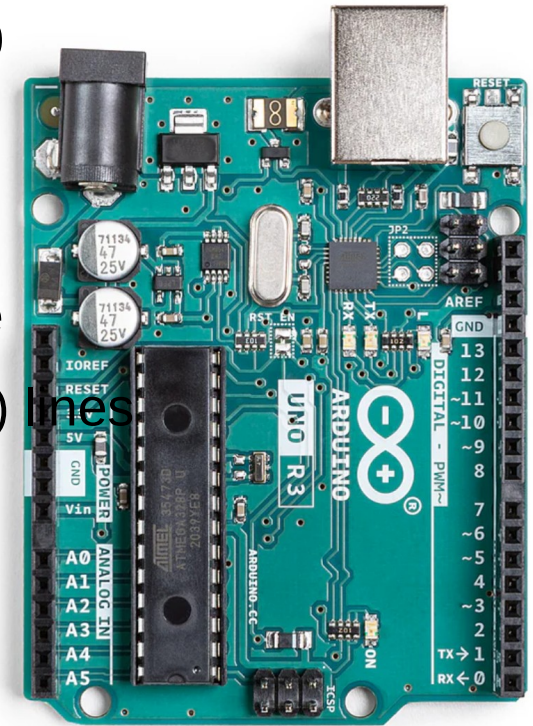
A brief introduction

By

Roger Walker

What is an Arduino?

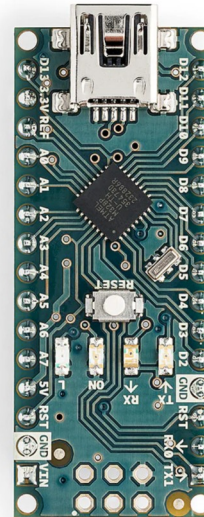
- Low cost development board for the ATMEGA328P microprocessor
- Target was US\$50, now about US\$20, clones under US\$10
- Uses a USB connection for programming/debugging
- Has a simple to use IDE (Integrated Editor/Debugger)
- Different inputs/output (I/O) lines, and form factors available
- Original has 14 digital I/O lines, 6 analogue in (or digital out) lines
- A single pin has enough power to light an LED
- The IDE can be used with all of the official Arduino boards



Why use an Arduino

(Instead of a Pi nano/Pi Zero/Full Pi/ESP32/Whatever)

- Cheap – Nano Clones are £2...3 from Aliexpress, ~£5 in UK
- Low power – Can be run from a few AA batteries / USB
- Fast boot time - can be under 1 second
- Modules are easy to use with prototyping boards – Easy to wire
- Lots of example wiring and code on-line – Lots of starting points
- Widely used - 50M+ downloads of the IDE, so lots of forums
- Open source hardware and software, that is well supported – Not locked in
- Lots of libraries available to interface to a wide range of devices – Easy to use
- Robust – Pretty tolerant of voltage errors, overloads, and other basic mistakes



Some programming hints

Simple programming #1

```
// The setup function does things once at power on
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// The loop function runs things over and over until turned off
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);    // Turn LED on (HIGH = On)
    delay(1000);                        // wait for 1000 ms (1 sec)
    digitalWrite(LED_BUILTIN, LOW);     // Turn LED off (LOW = Off)
    delay(1000);                        // wait for a second
}
```

- “;” separates commands, “//” starts comments, “{” and “}” enclose blocks of code, names are case sensitive
- Functions are blocks of code that can take parameters, do something, and possibly return a value (void = no value).
- setup() is called once when the power is turned on. It’s generally used to set the I/O pin directions, and the initial state of things
- loop() is then called repeatedly

Simple programming #2

```
// The setup function does things once at power on
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// The loop function runs things over and over until turned off
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);    // Turn LED on (HIGH = On)
    delay(1000);                        // wait for 1000 ms (1 sec)
    digitalWrite(LED_BUILTIN, LOW);    // Turn LED off (LOW = Off)
    delay(1000);                        // wait for a second
}
```

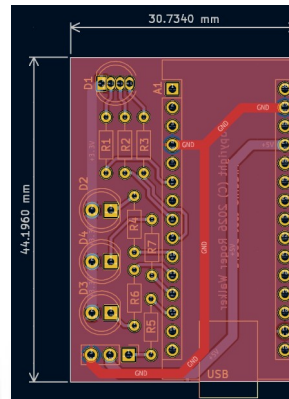
- **pinMode:** Takes the arguments “I/O PIN ID”, and “Direction”.
`pinMode(LED_BUILTIN, OUTPUT);` sets the PIN “LED_BUILTIN” to be an “OUTPUT”.
- **digitalWrite:** Takes the arguments, “I/O PIN ID”, and “State”.
`digitalWrite(LED_BUILTIN, HIGH);` sets the output voltage of PIN “LED_BUILTIN” to be “HIGH” (so 3.3...5V).
`digitalWrite(LED_BUILTIN, LOW);` sets the output voltage of PIN “LED_BUILTIN” to be “LOW” (so 0.0...0.6V).
- **delay:** Takes the argument of “Time”. This is the wait time in milli-seconds.



Let's start!

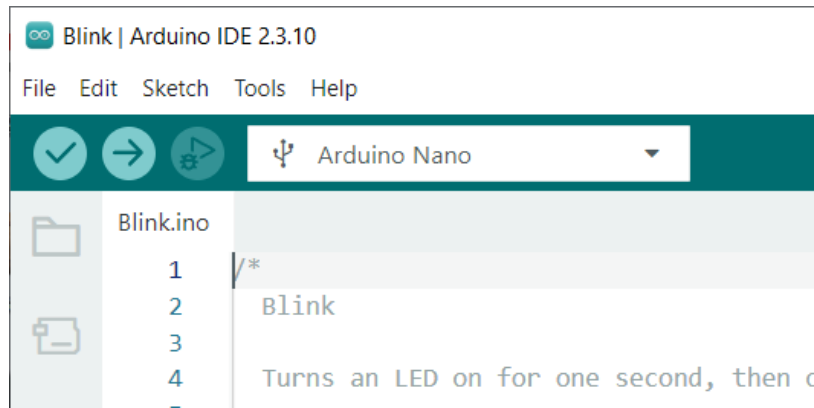
What should you have

- Windows laptop with the IDE pre-installed
- An Arduino Nano – The brain
- A simple test board connecting some of the output pins to simple devices
 - 3 of the pins to a Red+Green+Blue LED so we can mix colours
 - 3 of the pins to separate LEDs so we can animate them
 - 1 of the pins to a servo header, so we can connect a radio control servo
- A USB cable to connect them

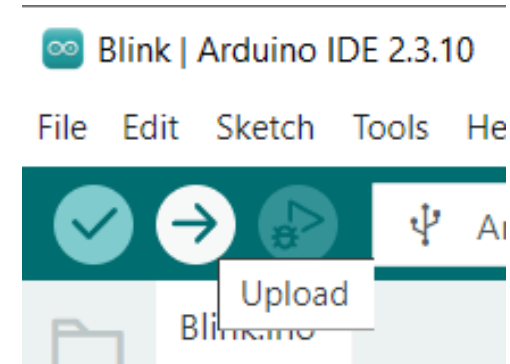


Starting things

- We start by launching the IDE, double click on this icon
- The IDE should start up, looking like this

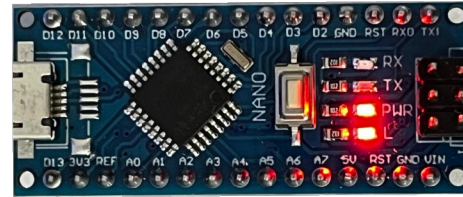
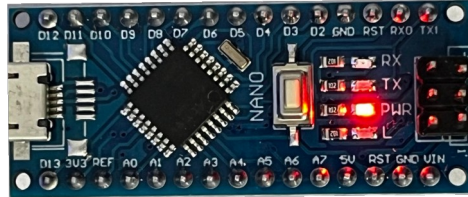


- Click on the “Upload” button
- You should see the following messages:
 - Compiling sketch
 - Uploading Sketch
 - Upload done



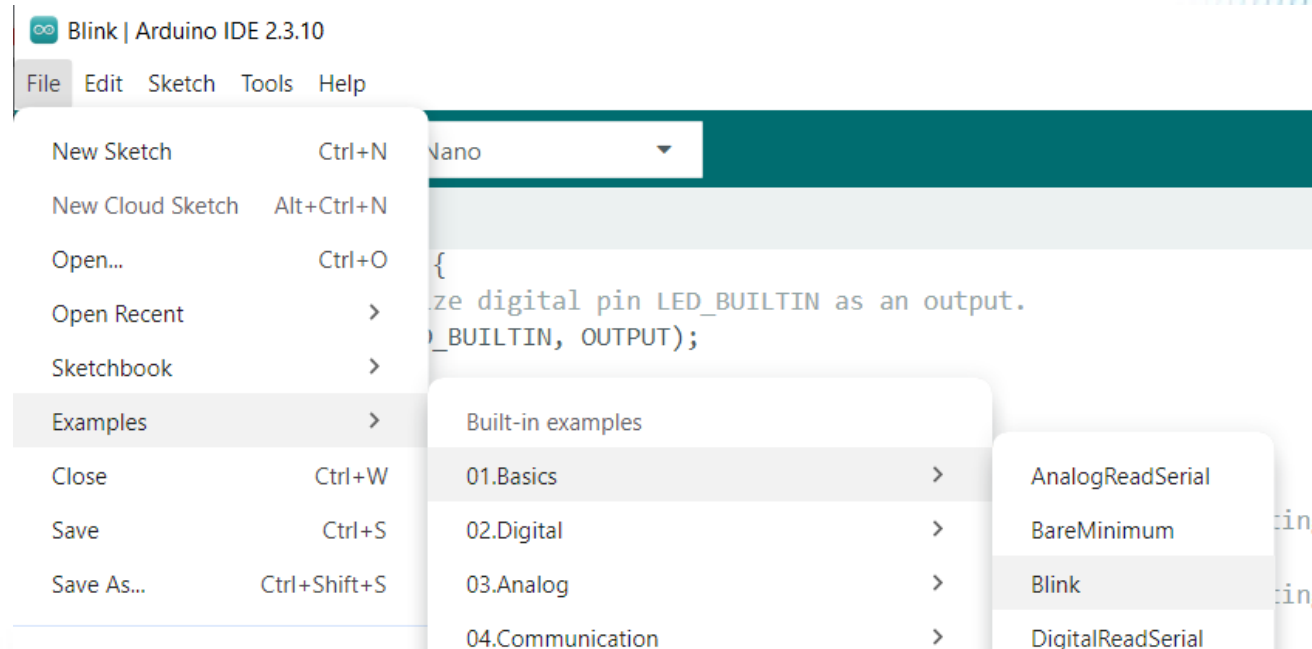
First result

- The on board LED marked “L” should be flashing on-and-off with a 1000 milli-second (=1 second), delay between the on and off events.



- This program is in the built in examples

- It's a simple test to make sure everything is connected before we try anything more complex.





First Edit!

Simple coding

```
// The loop function runs things over and over until turned off
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);    // Turn LED on (HIGH = On)
  delay(1000);                        // wait for 1000 ms (1 sec)
  digitalWrite(LED_BUILTIN, LOW);     // Turn LED off (LOW = Off)
  delay(1000);                        // wait for 1000 ms (1 sec)

  digitalWrite(LED_BUILTIN, HIGH);    // Copy of above
  delay(100);                         // Copy of above, change 1000 to 100
  digitalWrite(LED_BUILTIN, LOW);     // Copy of above
  delay(1000);                        // Copy of above
}
```

- So lets change the code to do something different
- Copy the first four lines of the loop() section, paste them before the “}”
- Change delay after the copied HIGH line from 1000 to 100
- The click on the “Upload” () button
- It should go through the compile/upload/done sequence
- You should see the LED do a long flash, then a short flash



Let's make some LEDs flash

Common LED board setup

To save time we will load this from a file

Use: File / Open / LED-Board-1 / LED-board-1.ino

```
// These are the pin definitions, all of the LEDs are low side switched
#define LED_ON HIGH
#define LED_OFF LOW

#define LED_1 7
#define LED_2 8
#define LED_3 9

#define LED_R 3
#define LED_G 5
#define LED_B 6

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize LED pins as outputs
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(LED_1, OUTPUT);
    pinMode(LED_2, OUTPUT);
    pinMode(LED_3, OUTPUT);
    pinMode(LED_R, OUTPUT);
    pinMode(LED_G, OUTPUT);
    pinMode(LED_B, OUTPUT);
}
```

Simple LED board loop

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_1, LED_ON);
  delay(500);
  digitalWrite(LED_1, LED_OFF);

  digitalWrite(LED_2, LED_ON);
  delay(500);
  digitalWrite(LED_2, LED_OFF);

  digitalWrite(LED_3, LED_ON);
  delay(500);
  digitalWrite(LED_3, LED_OFF);
}
```

- Use the “Upload” button, to compile the code, and program the board
- We should be able to see the separate LEDs, they are turning on for 500ms (0.5sec), then off
- So this lights them in the 1, 2, 3 sequence that keeps repeating

Changing the sequence

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_1, LED_ON);
  delay(500);
  digitalWrite(LED_1, LED_OFF);

  digitalWrite(LED_2, LED_ON);
  delay(500);
  digitalWrite(LED_2, LED_OFF);

  digitalWrite(LED_3, LED_ON);
  delay(500);
  digitalWrite(LED_3, LED_OFF);

  digitalWrite(LED_2, LED_ON); // Copy the LED 2 section from above
  delay(500);                  // and paste it here
  digitalWrite(LED_2, LED_OFF)
}
```

- Copy the LED_2 section to the end of the loop function
- Use the upload button, to program the board
- This time the sequence is 1,2,3,2, repeating as 1,2,3,2,1,2,3,2
- This zig-zag sequence is often known as the “knight rider” sequence (internet search “Kitt lights”)



Let's make some LEDs glow

Dimming LEDs

To save time we will load this from a file

Use: File / Open / LED-Board-2 / LED-board-2.ino

Note: The defines and setup() are the same as the previous code

```
// the loop function runs over and over again forever
void loop() {
    analogWrite(LED_1, random(20, 235));
    delay(500);
}
```

- analogueWrite: Takes the arguments “I/O PIN ID”, and “period”. Where “period” is 0...255. This turns the selected output pin on and off really quickly (Usually 490 or 980 times per second). Varying from 0 (0% on), through 127 (about 50% on), to 255 (100% on).
- Random: Takes a “min” and a “max” value to return, so `random(20, 235)`; returns a value between 20 and 235 (inclusive) [It's actually pseudo random, but fine for flickering an LED]
- So this code, sets the on time to $(20...235)/255$ (so about 8% to 92% on)
- Then leaves it “dimmed” for 500ms (0.5 seconds)

Flickering LEDs

- ```
// the loop function runs over and over again forever
void loop() {
 analogWrite(LED_1, random(20, 235));
 analogWrite(LED_2, random(20, 235)); // Copy the LED_1 line for LED_2
 analogWrite(LED_3, random(20, 235)); // Copy the LED_1 line for LED_3
 delay(50); // Change the 500 to 50
}
```

- If we change the brightness levels quickly then they appear to flicker
- This changes the 3 LEDs to different random levels every 50 milli-seconds (0.05 seconds), so 20 times per second.

# Questions & Answers